Students: Nathan Johnson, Erin Jones, Craig Stahlberg
Faculty Advisors: Javier Perez-Alvaro, Emily Stone

Incorporating Machine Learning with Cloud Based HVAC Monitoring Systems

Introduction

As a final step in the M598-498 University of Montana Capstone, the team tested several machine learning and data science algorithms with the data given at the beginning of the semester. The goal of this exploration was to practice real world implementation of learned data science models while determining if initial results from any of the selected methods produced successful detection of anomalies within the dataset. Anomaly detection within the machine sensor data given to use posed obvious use cases when the exploration began, from real-time detection of machine inefficiencies and excessive energy use to the possibility of predictive mechanical failure.

In the following sections you will find:

- An outline of feature extraction and treatment of the data
- Summaries of a variety of machine learning and data science techniques, including:
    - Random Forest Modeling
    - Median Absolute Deviation from the Median (MAD)
    - Agglomerative Hierarchical Clustering
    - Archetypes Analysis
- An exploration of the process of building simulated data
- Overall Limitations, Variables to Measure, and Conclusion

Conclusions are discussed in detail in the limitations and potential use cases and suggestions, however success came largely from examining data sets pre and post-treatment by PowerTron. The lack of labelled data rendered the training of supervised models and assessment of results difficult. The team was able to explore whether algorithms could identify pre and post-treatment data in an effective fashion as the data was labeled as pre and post treatment. Possible success in identification of anomalies in the machine's behavior is seen with threshold analyses of certain features, however this model's lack of mechanism for confirming results produces immature findings that are difficult to verify.

Section I. Feature Extraction and Exploration

Feature Extraction

In determining whether an HVAC system is performing within spec using sensor data, the team had to first determine which features should be used to best characterize the device's performance. Feature extraction represents the process of choosing which criteria will be used to make this decision, aiding in any pre-treatment applied to the data.

Initial meetings and reports provided by PowerTron combined with general research on HVAC function narrowed the teams approach to a focus on kW/ton, COP, and EER as appropriate measures of device efficiency.

Capacity was also examined as it has an implied value related compressor function. Capacity can be considered as a measure of the ability of the device to provide a certain amount of cooling. Based on manufacturing standards, there are expected values for capacity when one compressor is functioning, two compressors are functioning, and so on. Also related to compressor function was developing an understanding of extracting the amount of time one compressor, two compressors, three compressors, and four compressors were running in a given time period. This can be designated by either the total number of seconds a device was running during a day or by producing the average cycle runtime, manifested as a percentage.

Outdoor air temperature has an expected relationship with certain key variables mentioned above, such as the kW/ton used by the machine. When the temperature outside increases, the amount of energy required to provide cooling capabilities increases.

Included in the determination of which features to extract is the exploration of statistical representations of each measure over a given time period. Certain measurements are less impacted by sensor errors or expected spikes (outliers) in calculated measurements due to the common nature of equipment function.

For instance, kW/ton is expected to spike when a device cycles and compressors turn on or off. When a device turns on, it's capacity is close to zero, so the calculation results in the kW being divided by a very small number, which results in a very large kW/ton measurement. An effective statistical approach to handling spikes in calculated measurements is to take the median, as opposed to the mean (average), as it is less impacted by outliers in very large data sets. Hence, several features explored include the median kW/ton, COP, and the EER.

The variability of the data provided was also assessed during feature extraction. Typical statistics for the assessment of variability consist of the range, the interquartile range (the lower and upper boundaries of the middle 50% of the data), and the standard deviation. Unfortunately, the mean and standard deviation (based on the mean) are impacted by 'common' outliers in this data set and were discarded. The range, which reflects the minimum and maximum values in a given data set, was used. The interquartile range was also used to calculate a useful feature known as "percentage of outliers in a day" (calculated by dividing the number of datum categorized as outliers by the number of points in the entire set under examination).

Outliers indicate unusual values, and a point is classified as an outlier if it is further than some defined distance away from the median, known as thresholding. This is a categorical feature, which allows for determining whether a datum is or is not an outlier. Related to outliers and the creation of thresholds, the median absolute deviation from the median (MAD) was calculated and used as a feature. Much like standard deviation, which portrays the average distance from the mean, MAD illustrates the typical distance from the median of the data.

Finally, by exploring the relationship between data points, one can produce additional features based on the identified relationships. Particularly fruitful was the examination of the relationship between outside air temperature and average cycle runtime. The expected relationship would model an increase in cycle runtime with increases in temperature i.e. the device has to run for longer if it is

hotter outside. Preliminary results showed that runtime seemed to increase linearly with outside air temperature.
By demonstrating a linear relationship between these two variables, linear regression allows for the creation an equation to model this relationship. This equation was used to create a feature based on the deviation from the predicted relationship, where the deviation is referred to as an error. If the error was consistently large, this would merit attention, as it could mean that the machine was not performing as expected.

Each of the sub-sections under Section II: Deep Dive Methods, references the specific pre-treatment of data and the features extracted for each model tested. Feature extraction represents a key piece in any data science or machine learning implementation, as its impacts can range from the actual determination of which technique to explore to the actual success of the technique itself.

The team believes that further features could have been explored in future exploration specifically related to the features and variables that one would need to assess for successful predictive mechanical failure.

Section II. Deep Dive Methods

The following subsections outline the methods explored further by the team given time constraints. Each section produced different results and conclusions surrounding the effectiveness of any of the given algorithms or methods are independent of one another. Common limitations and conclusions resulting from all deep dives are discussed in the final section of this paper.

The Random Forest Algorithm

Method Description

The random rorest is a classification algorithm that will take a large set of inputs and put them into different categories. It is a "supervised" algorithm, which means that there must be a training set of data which allows for the machine to categorize future data sets based on given sample inputs and outputs. This is called labeled data. Consider a data set gathered by surveying people on whether they would buy a certain cell phone.

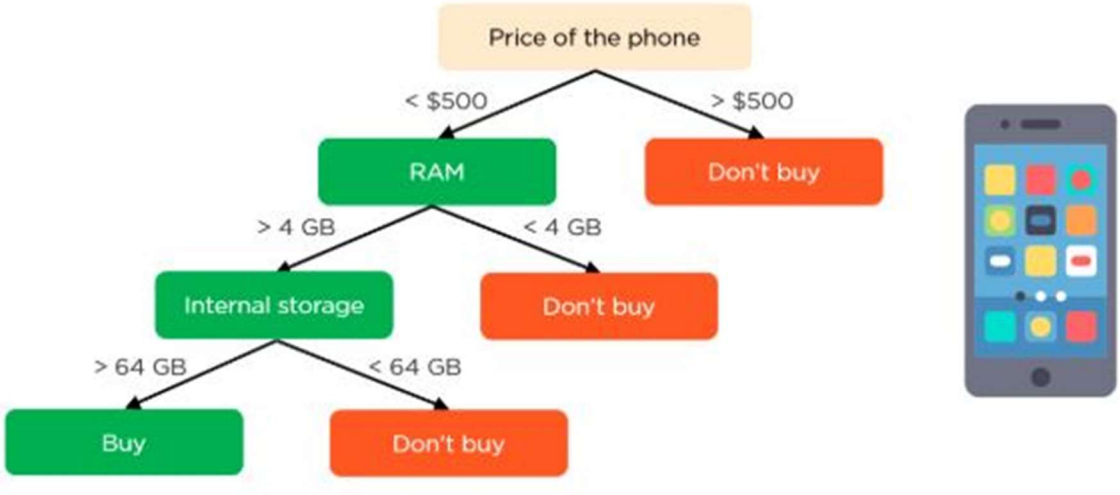
Phone	Price	RAM	Internal Storage	Label
A	$700	32 GB	128 GB	Don't Buy
B	$350	16 GB	128 GB	Buy
C	$90	1 GB	4 GB	Don't buy
…	…	…	…	…

Each input is a row representing a phone and can be thought of as a single data point with three features: price, RAM and internal storage. The key desired output is whether someone would buy

this phone, so each row is labeled "buy" or "don't buy."

The basis for a machine running a random forest algorithm is the use of decision trees. Figure 1 represents an example of a simple decision tree based on the above data.[1]

Figure 1 - Simple Decision Tree



Decision trees are fast, easy to understand, and work well with both non-numeric data (e.g., color, brand) and numerical data (price, internal storage etc.). However, decision trees carry certain limitations. They can be inflexible; for example, in the tree above, there is no option to allow for a phone more than $500 to be purchased. This inflexibility produces a bias in the characterization of the data, though it may have simply occurred because no respondents represented in the provided training set wanted to buy an expensive phone. Additionally, decision trees are very sensitive to small differences in the training data provided

The random forest algorithm improves on decision trees, because it is an ensemble method. This means it uses many methods or trials to make many (ideally) independent predictions and combines them to form one prediction. The intuition here is that having many independent predictions reduces the bias that can come from the training set (the one above is biased against expensive phones) and copes better with the natural variability found in the real world.

The random forest algorithm uses smaller decision trees and then determines which outcome was most common for its output.

In the example above, the smaller decision trees could be formed by random subsets of the original tree. For example, one tree might be just a one node tree (Figure 2), that is one level deep; another might be a two node, two level deep tree (Figure 3); the final subset represents a third tree (Figure 4) that was included in Figure 3, as duplication of nodes between the trees poses no issue.

[1] https://www.simplilearn.com/ice9/free_resources_article_thumb/phone-price.JPG

Figure 2 - Single Node, Single Level Decision Tree

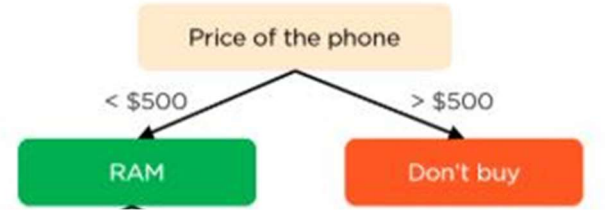


Figure 3 - Decision Tree with Two Nodes & Two Levels

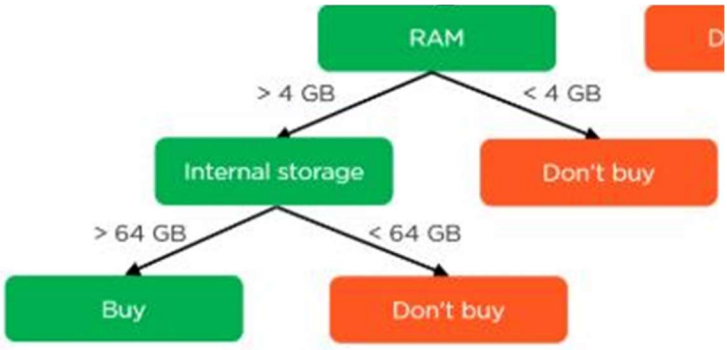
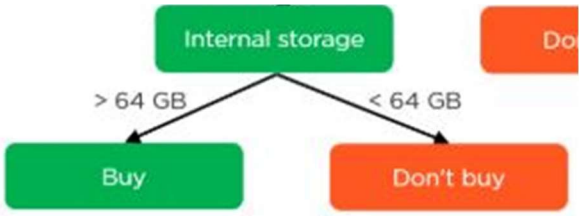


Figure 4 - Duplicate Decision Tree



Each tree in the ensemble can be considered a separate 'vote' for a data point to be placed in the 'buy' or 'don't buy' labeling category. Using the above ensemble of trees, a phone > $500 could be bought if it had > 64 GB of storage and 4 GB of RAM. The first tree would push the needle towards not buying the phone as the price is too high, but the latter two trees would push the tree towards buying it. The ensemble classification results in a "vote" of two to one, thus the phone would be bought and the bias from the simple decision tree is removed.

Random forest has several other advantages that improve its performance over simplified decision trees. For example, small trees could also be trained on different datasets, further increasing flexibility. Also, the relative importance of different features can be analyzed by comparing the performance of different tree configurations to the training dataset.

For the purposes of this project, the random forest was trained on 39 days of Macy's data. Nine days fall into the category of pre-treatment in October, 2019 (these were labeled anomalous) and 30 from the category of post-treatment in May, 2020 (these were labeled normal). The features used were:

- Both median and median absolute deviation from the median (MAD) for:
    - kW/Ton
    - COP
    - EER
- average and maximum outside temperature
- median and MAD capacity (TON) stratified by the number of compressors running- 1, 2, 3 or 4
- the time 1, 2, 3, and 4 compressors are turned on

The data was filtered in two ways. First, data points were only considered when Compressor 1 was on. In the data set considered, there were four compressors labeled 1 – 4 that always came on in order. Generally, Compressor 1 ran most frequently during the day; while Compressor 2 would periodically kick in to provide extra cooling capacity. Compressors 3 and 4 rarely ran. Second, data points were only considered when CFM was at least 100 cubic feet per minute. This helped to remove spikes in the calculations that occur when the machine is coming on or turning off, as CFM drops to zero at these times.

Figures 5 and 6 below represent examples of the decision trees the random forest model generated from the pre-treated data for the selected dates.

Figure 5 - Random Forest Macy's Example 1

Decision tree 1 generated on 39 days of Macy's data (dates from 10-1-19 to 10-8-19 and 5-1-20 to 5-31-20)

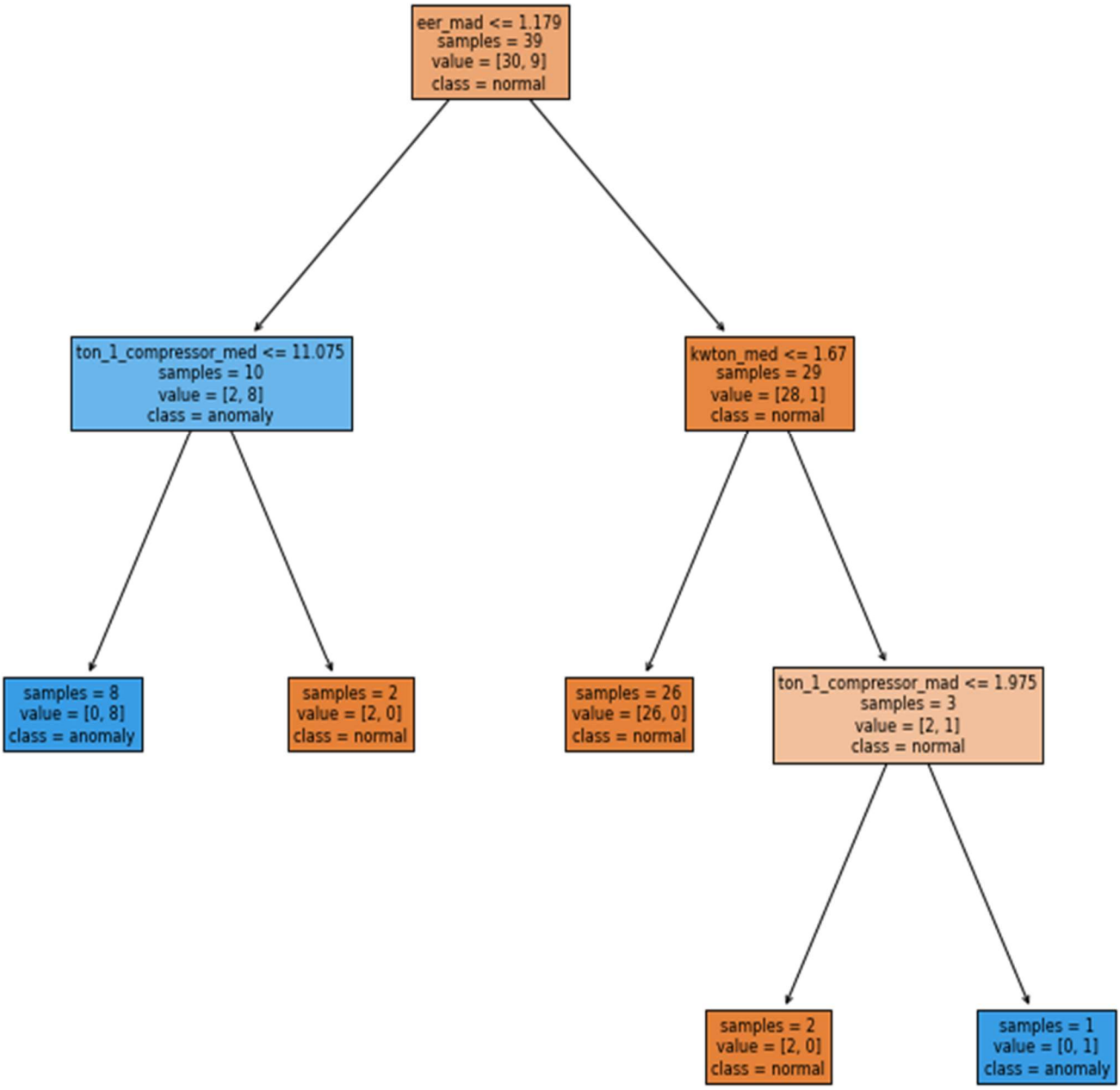
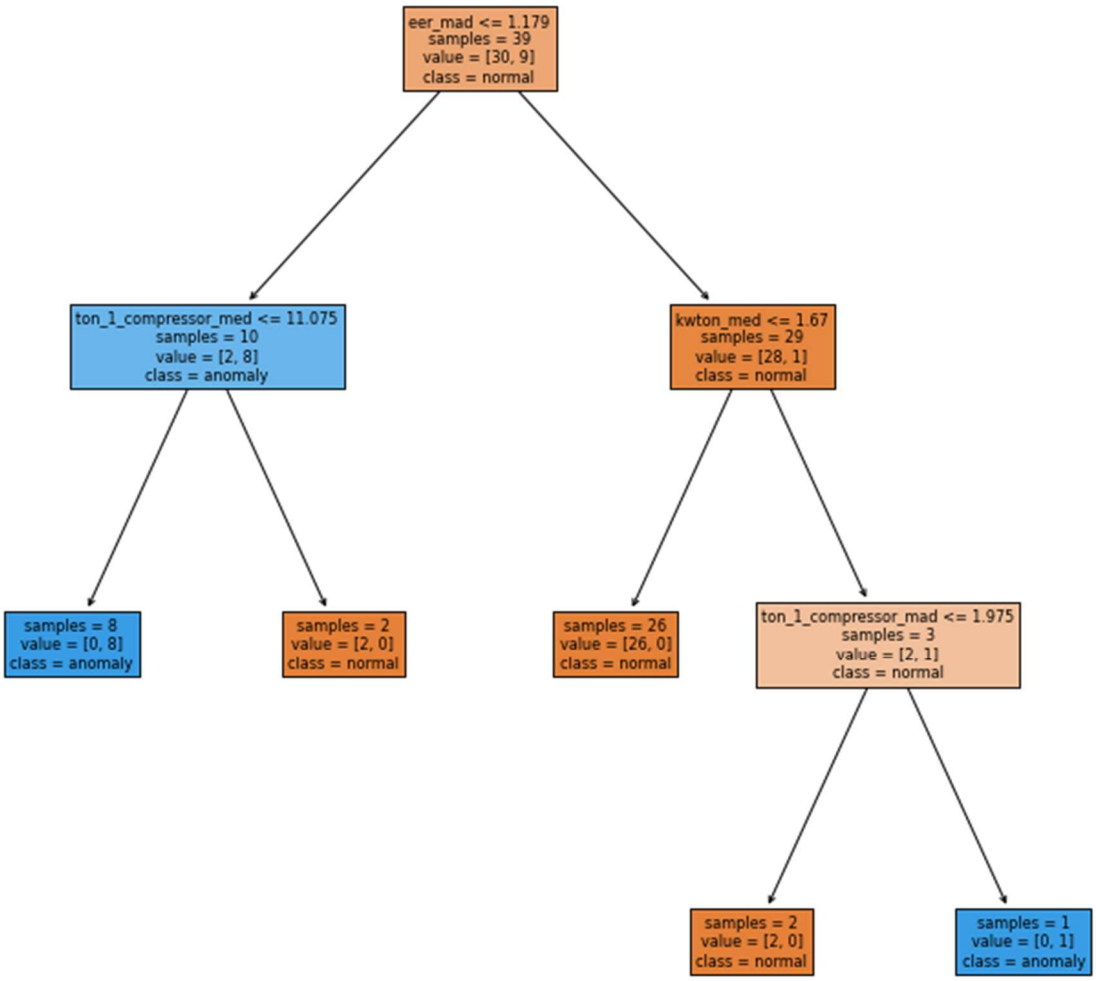


Figure 6 - Random Forest Macy's Example 2

Decision tree 2 generated on 39 days of Macy's data (dates from 10-1-19 to 10-8-19 and 5-1-20 to 5-31-20)



Results

Typically, another labelled data set is used to test a supervised machine learning algorithm. This data set will be used as input for the algorithm and the output will be compared to the known labels, allowing an accuracy percentage to be calculated. It follows that without more labelled data, it is not possible to conclusively test performance and produce precision accuracy measurements.

While lacking an understanding of true method accuracy serves as a notable limitation stemming from an absence of labeled data, the team was able to mimic a testing data set by using pre and post-treatment data as ‘anomalies’. Once trained on the labeled data from the 39 days in October 2019 and May 2020, the algorithm was applied to Macy's data from June 2020 to November 2020 and the following days were classified as anomalies:

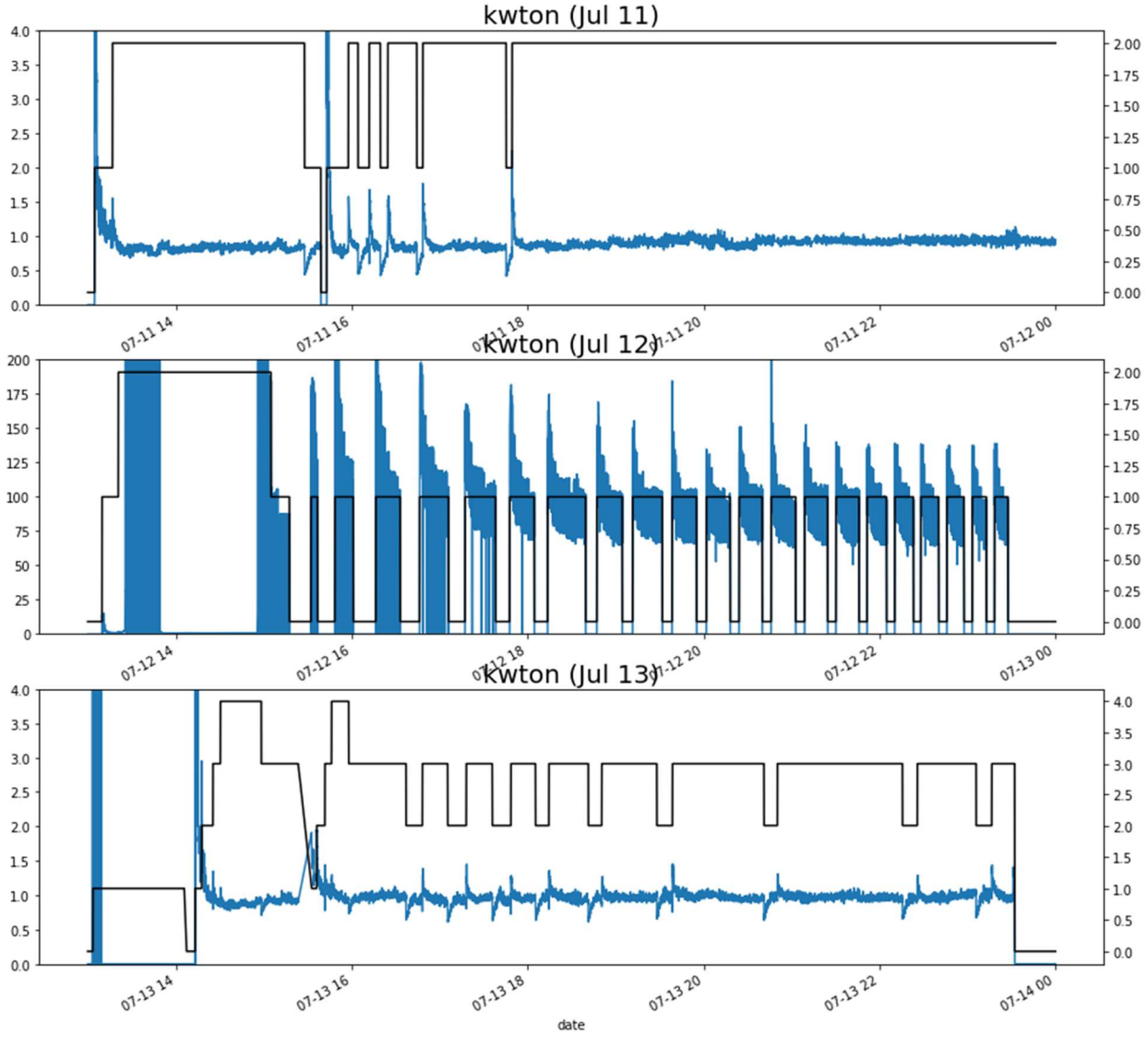
- 2020-07-12
- 2020-09-01
- 2020-09-09

- 2020-09-10
- 2020-10-19
- 2020-10-25
- 2020-10-26
- 2020-10-27
- 2020-10-28
- 2020-10-29
- 2020-10-30
- 2020-11-01
- 2020-11-02
- 2020-11-10
- 2020-11-16
- 2020-11-22
- 2020-11-23
- 2020-11-27

Further investigation into the dates listed shows anomalies, so the algorithm is at least partially successful in the identification of anomalies. For instance, consider the first anomaly July 12, 2020. Plotting kW/ton for that day and the surrounding days which were not flagged as anomalous, produces the results shown in Figure 7.

Figure 7 - July 11th - 13th 2020 Plots

The left vertical axis represents kW/Ton (blue values), while the vertical axis on the right represents the number of compressors running (black line). The first graph represents the 11th, the second the 12th and the third the 13th.



We can see the values for July 12th (middle graph) have a visually distinct pattern and are about 100 times higher. The values are enough higher that the scale is modified automatically by the algorithm to show peaks of even over 200, while the 11th and the 13th both hover around 1.5 and 2.

As a secondary result of this method, the random forest algorithm produces a list of features ranked by importance as they relate to the classification being performed by the algorithm. This list is shown in Figure 8.

Figure 8 - Features by Importance

	feature_importance
eer_mad	0.146297
cop_med	0.140376
eer_med	0.113870
time_1	0.095750
ton_1_compressor_mad	0.091308
kwton_med	0.074798
ton_2_compressors_med	0.063031
cop_mad	0.051234
ton_1_compressor_med	0.049846
ton_2_compressors_mad	0.048344
avg_temp	0.034865
kwton_mad	0.031731
max_temp	0.021741
time_2	0.019394
ton_3_compressors_mad	0.011990
ton_4_compressors_med	0.004021
time_3	0.001073
ton_3_compressors_med	0.000331
ton_4_compressors_mad	0.000000

The more ‘important’ a feature is, the more it impacts whether the algorithm distinguishes a day to be ‘normal’ or ‘anomalous. This can be useful information in selecting features to report and investigate further.

Conclusion and Possible Uses

In conclusion, this method shows promise but needs more labeled data to fully evaluate. If PowerTron were able to provide a labeled data set with clear indicators of instances where an anomaly they were interested in tracking occurred, this algorithm could be used to investigate anomalies non-labeled datasets from other months of operation.

Furthermore, when used in combination with other statistical models, it may be possible to create a deeper understanding of 'warning signs' that typify pre-breakdown behavior in a device. With this understanding, a labeled dataset could be created that trains the algorithm to detect pre-breakdown behavior and flag this occurrence in an active manner.

Additionally, impacts of decisions made by building engineers and managements could be tracked. The training dataset would need labels for variable measurements known to be an outcome of a user behavior i.e. When building management decides to do x action, the COP/EER/etc do y. The algorithm would then identify and track these incidents over time.

Obvious limitations to this method come rooted in the fact that it is a supervised learning algorithm requiring a training set of labeled data to produce results, thus requiring knowledge of what characterizes an anomaly prior to running the algorithm. Further findings and uses could possibly be generated via additional labeled data or data annotations describing decisions made by building engineers, mechanical issues, and more.

Detecting outliers – An Analysis of Variability:

Median Absolute Deviation from the Median (MAD) Method Description

The Median Absolute Deviation from the Median (MAD) is a measure of variability. This idea of MAD was introduced in a talk by the head data scientist at Datadog.[2] The method is more simply implemented than the interquartile range and is less impacted by outliers by using the median. While the values in any data set are expected to vary, if they change 'too much' it should trigger further investigation within an analysis. Defining 'too much change' within a dataset is commonly called "thresholding." Thresholds are often set manually, but the team wanted to investigate if these thresholds could be automatically applied. If a machine could automatically and actively apply thresholds to a dataset characterizing machine behavior, the use cases would be extensive.

The milestone 1 report for this project discussed using the interquartile range (IQR) to identify outliers. Presented below is a similar method using a different statistic: the median absolute deviation from the median.

MAD is calculated as follows:

Find the median of the dataset D = { 1, 2, 3, 4, 5, 6, 100}

[2] Detecting outliers and anomalies in real-time at Datadog, Homin Lee. Presented at OSCON, Austin Texas, May 16-19, 2016. https://www.youtube.com/watch?v=mG4ZpEhRKHA

	median =4
Find the deviations from the median	deviations = { -3, -2, -1, 0, 1, 2, 96 }
Take the absolute value	absolute deviations = { 3, 2, 1, 0, 1, 2, 96 }
Find the median of the absolute deviations	sorted absolute deviations = { 0, 1, 1, 2, 2, 3, 96 } **MAD = 2**

Using the MAD, thresholds can be set: median +/- MAD * tolerance factor. In the above data set, using a tolerance factor of 3, our thresholds would be set at -2 and 10 (4 +/- 2*3). Thus, any values below -2 or above 10 would be considered outliers.

Results and Possible Use Cases

Consider an example from the Macy's data set during the month of May 2020. The data was filtered so that only values where compressor 1 was on and cfm > 100 were considered. (Figures 9 and 10). The median kW/ton in the May 2020 dataset was 1.09, while the MAD was 0.19. On May 4, 2020, 1.85% of the data were outliers (Figure 9), but on May 5th, 52.98% of the values were outliers (Figure 10). This breach in the threshold could indicate a malfunction and warrant further investigation.

Figure 9 - kW/Ton Outliers in May 2020 (Macy's)

Outliers in the kW/Ton measurements are marked with a red dot; outliers in the CFM measurements are notated with a green dot, while blue marks standard kW/Ton measurements over time.

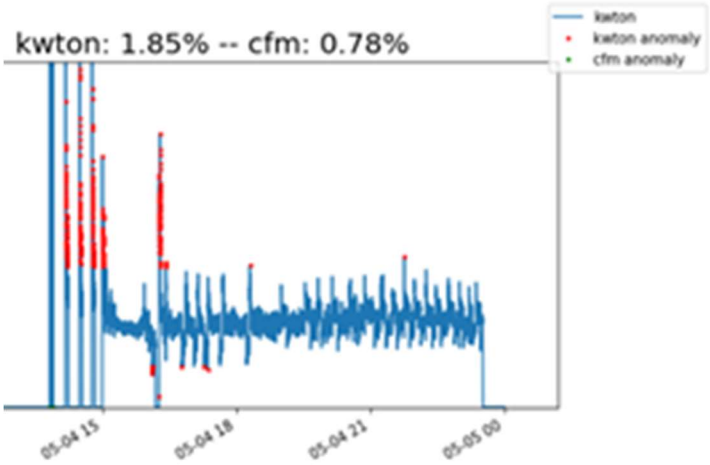
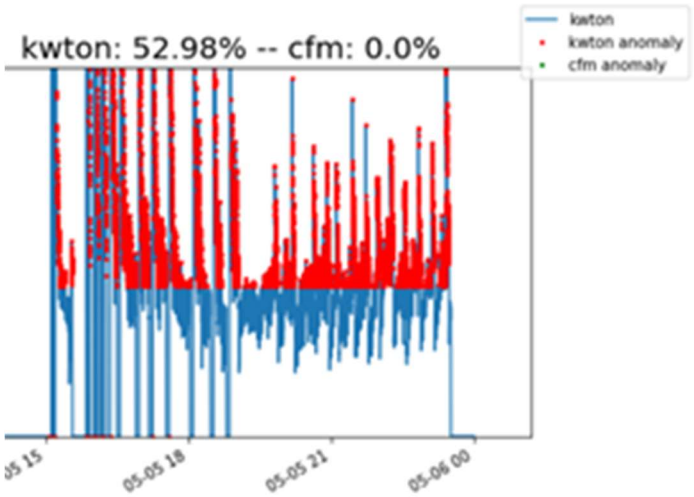


Figure 10 - kW/Ton Outliers in May 2020 (Macy's)



To apply this method, one must choose the tolerance factor, the period of time for which the median is calculated, and how many outliers during a period would trigger an alert. For this exploration, the team selected a tolerance factor of 3, calculated the median in monthly increments, and specifically triggered the alert when the amount of outliers contained in the dataset amounted to 30% or more of the data within the set.

Depending on the expected regularity of the data, one could 'zoom in' by reducing the threshold for percentage of the set classified to be outliers. If changes occur that may be of interest within smaller periods of time, the time increment for median calculation could be modified to ensure changes shifts in the median over time are not lost. To understand the best measurement conditions, selected presets for the algorithm should be tested via experiments on historical data to determine what provides the most pointed outcomes. For example, median could be a running value calculated continuously, or could be calculated only on data taken during a matching outdoor air temperature.

This model has potential to detect possible mechanical failure in real time, however more historical data would be needed to examine trends and set appropriate thresholds based on understanding of the data in previously recorded mechanical failure or malfunction. This use case may also need further variables not being calculated at this time, such as pressure, that may need more sensor equipment. Historical data would need to be labeled so results can be understood against real record of mechanical failure.

Another key statistic that would be worth pulling on an ongoing basis to better assess this methodology would be a running measure of the spread of the data.

This method has real potential use cases within its ability to be connected with an application and actively flagging the engineer or device user when an amount above the threshold of outliers is detected.

Agglomerative Hierarchical Clustering

Overview

Hierarchical clustering is a type of clustering algorithm that takes input data and categorizes it into groups known as clusters. To cluster data, the algorithm's input data is assembled as objects, each object being characterized by a series of variables. In clustering time series, measurement variables are selected as features to represent a particular measurement period, each measurement period representing an object to be clustered by the algorithm, grouped according to the similarity of its features against other objects.

For this exploration, pre-treatment of the data involved selecting an amount of time and then extracting a handful of features to characterize that period of time. Ideally, the clusters produced by the algorithm would classify the objects and produce groupings to show which objects were anomalies and which represented those objects with standard measurements.

The team decided to build a data set of days (objects) that each had seven variable measurements the algorithm then used to compare one day to the next. Characterizing each day were kW/Ton, median COP, capacity in tons BTU when compressor 1 was running, capacity in tons BTU when both compressors 1 and 2 were running, maximum outside air temperature, the time 1 compressor was active, and the time when both compressors 1 and 2 were active. Thus, each day was defined as an object by the above variables and then compared to the other days (objects) in the data set based on the similarities and differences between the measured features of the given object.

For hierarchical clustering, the input data consisted of 8 days from the Macy's data set in October 2019 and 30 days from the Macy's data set in May 2020. The October data stood to characterize device performance prior to PowerTron's treatment of the device (pre-treatment) and the days in May portray device performance post-treatment. These days were selected due to the implied difference in the efficiency of the device pre and post-treatment that should be manifested in the feature measurements collected for these days. The algorithm should cluster the days accordingly and the success of the method's output can be compared to the expectation that these sets of days are different and should be classified as such.

Finally, an essential step for the pre-treatment of data is the normalizing of the input data set. If each row of data represents a day and the columns are the characteristics measured and recorded for that day, the features can be normalized to produce a percentage based on all other measurements for that feature. The differences between raw and normalized data can be seen in Figures 11 and 12 below.

Figure 11 - Raw Macy's Data

kW/Ton	Median COP	Capacity for Compressor 1	Capacity for Compressor 2	Outside Air Temp	Compressor 1 Active Time (Seconds)	Compressor 2 Active Time (Seconds)
1.6	2.131514441	9	16.26	98.78	30403	17645
1.62	2.118673993	8.82	16.43	100.22	34432	20778
1.67	2.105987322	9.17	16.39	101.3	32508	19823
1.57	2.170986931	9.18	16.06	96.08	30394	15391
1.61	2.170986931	9.54	16.41	100.4	32550	18945
1.65	2.131514441	9.15	15.88	102.38	29710	15231
1.71	2.056724461	7.78	12.92	82.94	19044	5076

Figure 12 - Normalized Data

kW/Ton	Median COP	Capacity for Compressor 1	Capacity for Compressor 2	Outside Air Temp	Compressor 1 Active Time (Seconds)	Compressor 2 Active Time (Seconds)
0.829015544	0.503030303	0.440744368	0.542	0.964836882	0.882986757	0.809143853
0.839378238	0.5	0.431929481	0.547666667	0.978902129	1	0.952813317
0.865284974	0.497005988	0.44906954	0.546333333	0.989451065	0.944121747	0.909020039
0.813471503	0.512345679	0.449559256	0.535333333	0.938464544	0.882725372	0.705782547
0.834196891	0.512345679	0.46718903	0.547	0.980660285	0.945341543	0.868757738
0.85492228	0.503030303	0.448090108	0.529333333	1	0.86286013	0.698445453
0.886010363	0.485380117	0.380999021	0.430666667	0.810119164	0.553090149	0.232769294

Figure 11 represents the data prior to normalizing the measurements, while Figure 12 shows the normalized characteristics. Normalization shifts and rescales the data points within a field (e.g. kW/Ton or Median COP) so they end in a scale from 0 to 1 and thus are more similar across the dataset[3]. The normalized point is generated by evaluating all data points within each feature to find a range, then taking the raw data point, subtracting the minimum data point within the field and dividing the result by the range for the field.

$$X_{normalized} = (X_{raw} - X_{minimum})/(X_{maximum} - X_{minimum})$$

The normalized data points illustrate the raw measurements in a controlled fashion, facilitating the visualization and comparison of the points within the set. This is important when building complex comparisons, such as the distance matrix seen in the next sub-section.

Basis of the Algorithm

The process used by the hierarchical clustering algorithm to compare and then cluster the data is what makes each algorithm unique. Hierarchical clustering builds a hierarchy based on the distance of each object from other objects in the multi-variate space. Consider mapping each object in a space with many dimensions (in this case, each day in a space with 7 dimensions) and then analyzing the distance between each point.

It is helpful to start by imagining a generic data set where each object is characterized by only two measurements i.e. each object has two features used to compare it to other objects in the set. The algorithm begins by creating a distance matrix, also referred to as a dissimilarity matrix, to calculate the distance between each object.

An example of the distance matrix produced for this generic, two-feature data set is shown in Figure 13 below[4].

[3] https://towardsdatascience.com/normalization-vs-standardization-explained-209e84d0f81e
[4] https://www.displayr.com/what-is-a-distance-matrix/

Figure 13 - Generic Distance Matrix Example

Raw Data

	X	Y
A	9	49
B	24	54
C	51	28
D	81	54
E	81	23
F	86	32

Distance Matrix

	A	B	C	D	E	F
A	0	16	47	72	77	79
B	16	0	37	57	65	66
C	47	37	0	40	30	35
D	72	57	40	0	31	23
E	77	65	30	31	0	10
F	79	66	35	23	10	0

The way that distance is calculated for each object (A-F in the figure above) is by using the Pythagorean theorem. For instance, the distance between objects A and C is calculated by the following formula...

$$\sqrt{(51-9)^2+(28-49)^2} = 46.95743\ldots \approx 47$$

This example helps to understand the process of creating a distance matrix in a simplified manner. However, in the analysis of the given data set, each object is characterized by 7 variables. In this case, distance is instead calculated using the Euclidean distance formula so all variables can be considered, as opposed to the two-variable distance produced by the Pythagorean theorem. The resulting distance matrix can be found in the 'Results' sub-section for hierarchical clustering.

Once the algorithm has produced a distance matrix, the distances are compared in a hierarchical fashion, first grouping the two data points closest together to form a cluster. The assumption that each data point is initially treated as an independent cluster and then iteratively grouped together using the distance matrix defines this method to be agglomerative, as opposed to divisive[5]. Once a link has been created between the closest two points, the next closest distance is determined between the two-point cluster and the other data points, working to find the shortest distance. After the next shortest distance is determined, a link is created. This process occurs iteratively and produces a visualization of the hierarchy mapping the distance between points known as a dendrogram. The dendrogram shows linkages across all data points and is eventually used to determine the input for number of clusters when commanding the algorithm to produce cluster assignments.

After the number of clusters has been determined by the dendrogram, the algorithm can be run set to the number of clusters determined to receive an output depicting the cluster assignments for all input objects e.g. Day x belongs in cluster y based on its relationship to all other data points. To better understand the cluster assignments and place them in context of the selected features, the objects can be visualized with a scatter plot. Each data point on the plot represents a day as it falls within the framework of the selected features – while the color of the points in the plot can be modulated to represent what cluster the point was assigned to.

[5] https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec

Scatter plots and a dendrogram for the Macy's data are included in the following results section.

Results, Conclusions, and Possible Uses

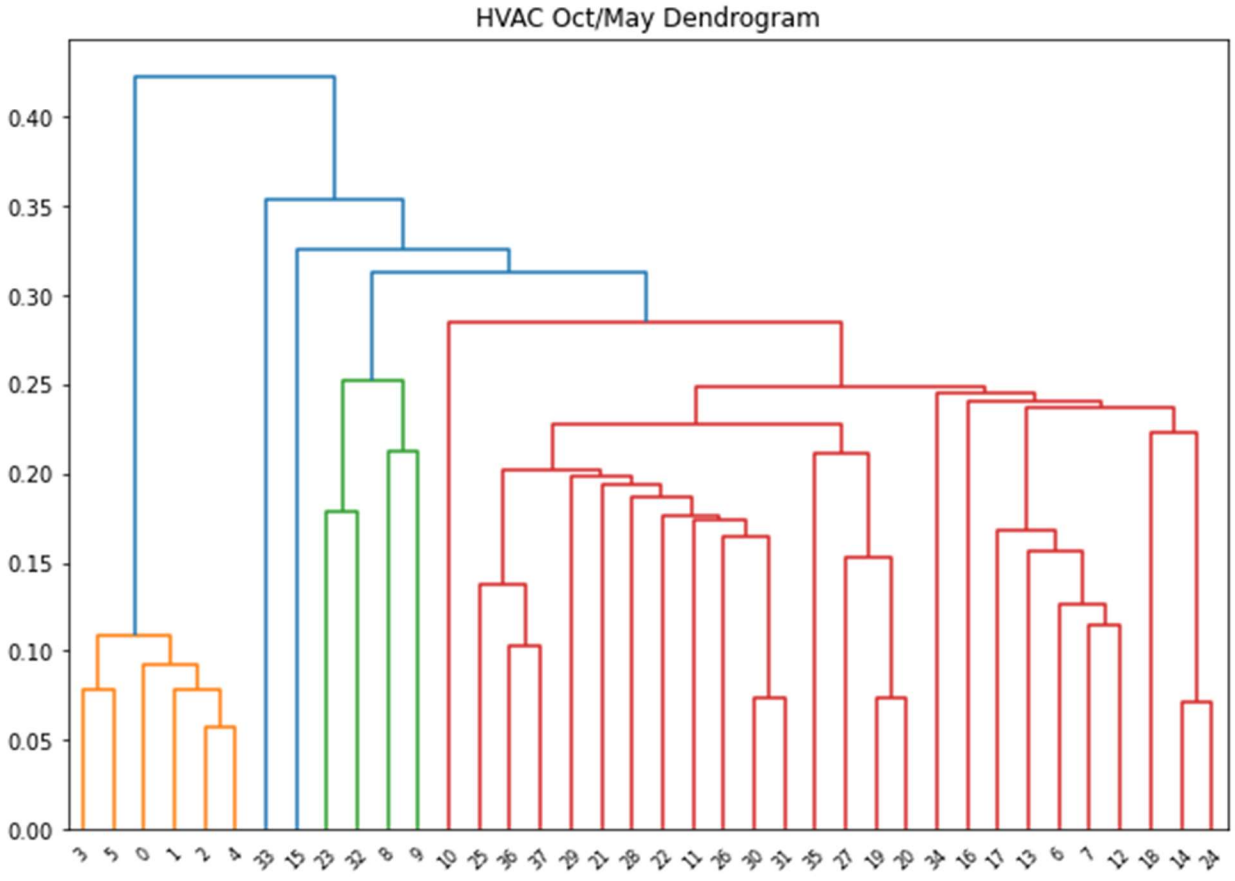
To begin, the team used the Euclidean distance and created a distance matrix between all points within the normalized data set. When calculated a 38 x 38 distance matrix was produced showing the distances between all 38 days assessed. A portion of this matrix can be seen in Figure 14.

Figure 14 - Macy's Oct 2019/May 2020 Dissimilarity Matrix

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0.186432901	0.125532657	0.108764278	0.092359393	0.121578061	0.696061443	0.722990475	1.028613686	1.121545263	0.972116257	0.736516169
2	0.186432901	0	0.078276387	0.278745821	0.107139725	0.291210493	0.874843247	0.905422898	1.211975131	1.306065683	1.085473066	0.765347196
3	0.125532657	0.078276387	0	0.225201688	0.056836203	0.226914133	0.812993105	0.840687951	1.148994977	1.241392478	1.048642374	0.753578938
4	0.108764278	0.278745821	0.225201688	0	0.182040216	0.077962649	0.608701979	0.637234102	0.945853621	1.046308183	0.922644476	0.755820939
5	0.092359393	0.107139725	0.056836203	0.182040216	0	0.193330495	0.782176879	0.80926225	1.109584451	1.207694149	1.003114323	0.729186231
6	0.121578061	0.291210493	0.226914133	0.077962649	0.193330495	0	0.603639612	0.622287105	0.939295079	1.03703551	0.939172588	0.778416383
7	0.696061443	0.874843247	0.812993105	0.608701979	0.782176879	0.603639612	0	0.126137878	0.509730938	0.574190696	0.985279552	1.124099542
8	0.722990475	0.905422898	0.840687951	0.637234102	0.80926225	0.622287105	0.126137878	0	0.436310316	0.500772406	0.947513975	1.114439431
9	1.028613686	1.211975131	1.148994977	0.945853621	1.109584451	0.939295079	0.509730938	0.436310316	0	0.212910903	0.804713917	1.159276063
10	1.121545263	1.306065683	1.241392478	1.046308183	1.207694149	1.03703551	0.574190696	0.500772406	0.212910903	0	0.944624155	1.258638056
11	0.972116257	1.085473066	1.048642374	0.922644476	1.003114323	0.939172588	0.985279552	0.947513975	0.804713917	0.944624155	0	0.599460657
12	0.736516169	0.765347196	0.753578938	0.755820939	0.729186231	0.778416383	1.124099542	1.114439431	1.159276063	1.258638056	0.599460657	0
13	0.720505541	0.898649863	0.8376177	0.629045626	0.803615689	0.617358513	0.13609007	0.114921393	0.474086648	0.565377775	0.972366971	1.145576544
14	0.652003587	0.835274631	0.770057137	0.568453344	0.736263179	0.559763679	0.15714872	0.166570238	0.460867841	0.555139007	0.892700597	1.034148383
15	0.75731376	0.925028806	0.872674428	0.671805537	0.824455999	0.684153898	0.497296077	0.464771562	0.396886541	0.566675772	0.523082188	0.835288498
16	1.059077285	1.214415349	1.157317621	0.98934524	1.137199301	0.974507244	0.508063059	0.54899306	0.77512984	0.795443921	1.415897397	1.565789942
17	0.880925346	1.05309576	0.989198907	0.804831224	0.966398762	0.789658359	0.24045596	0.2739239	0.576143833	0.587922319	1.185530304	1.336149636
18	0.578586588	0.763356579	0.698218619	0.505294531	0.665948269	0.502490348	0.245205797	0.249808717	0.494122909	0.565806277	0.820796725	0.908168115
19	0.764190696	0.944716031	0.886701228	0.6730705	0.843849205	0.678577953	0.305318601	0.281801814	0.313378396	0.462997105	0.745294197	0.999349239
20	0.873352821	1.00793822	0.970295997	0.797173621	0.918573792	0.821846331	0.769045893	0.743984746	0.637843652	0.802804777	0.316359835	0.745116555
21	0.901342147	1.039446881	0.999153237	0.827255639	0.948887348	0.847864311	0.779459389	0.745419525	0.612166873	0.76911983	0.285300209	0.747379486
22	0.721506437	0.806929348	0.779375989	0.694367647	0.737618503	0.717444364	0.938337565	0.918415369	0.915338357	1.042869379	0.320481598	0.321528961
23	0.917579084	0.973421067	0.952583761	0.909179785	0.91807216	0.933277933	1.170910776	1.158275616	1.119427008	1.241704263	0.425381585	0.315968603
24	0.995261812	1.174978033	1.116074233	0.910790682	1.077341605	0.913256919	0.496760537	0.436699408	0.252601275	0.277035796	0.786266688	1.112149056
25	0.714402417	0.883035774	0.831183622	0.626955317	0.783160159	0.643113184	0.463598476	0.440445323	0.426043894	0.580657279	0.542975948	0.817819813

From the matrix, the dendrogram is then produced, starting by grouping the closest two days, then adding the 3rd closest and so on until the iterative process has analyzed and assigned a cluster to each day. Figure 15 shows the dendrogram produced by the algorithm package when the October 2019/May 2020 dataset is inputted.

Figure 15 - Dendrogram of October 2019/May 2020 Data



The y-axis in figure 15 is used to measure the height of the line connecting each object notated on the x-axis. The taller the line representing the linkage between objects, the further apart those objects are calculated to be. The colors within the dendrogram indicate the cluster assignments and suggest that 4 clusters should be used when assigning clusters within the data.

At this point, all the inputs have been collected to run the hierarchical clustering algorithm that produces a matrix assigning a cluster (cluster 0, 1, 2 or 3) to all 38 days included in the dataset. The output matrix is seen in figure 16.

Figure 16 - Cluster Assignments within October 2019/May 2020 Dataset

```
Out[8]: array([1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0,
               0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0], dtype=int64)
```

The highlighted days in figure 16 are the first 8 days listed in the dataset i.e. the days from October 2019, the others listed are days in May 2020. There is some overlap in the measured device behavior on days 7 and 8, as they fell into the ‘0’ cluster where the majority of days in that cluster are days from May. This overlap is minimal as the remaining days from October were recognized as a cluster of their own, not containing any data sets from May 2020.

There were also two outliers identified within the days from May, clustered within clusters 2 and 3. These are seen to be outliers as they are single days that were dissimilar enough from the points in clusters 0 and 1 that the algorithm chose to classify them in their own categories entirely.

Overall, the result produced is reasonably successful, as the majority of days from the pre-treatment data were classified separately from the post-treatment data. This aligns with the expectation that the behavior of the device would be different between pre and post PermaFrost treatment implemented by PowerTron.

To examine possibilities even further, the team decided to look at several scatter plots of the data, where the clusters have been identified. The two most interesting plots produced that illustrate not only the behavior of the clusters, but also the success of PowerTron's treatment are shown below in figures 17 and 18.

Figure 17 - COP Against Outside Air Temperature

COP is measured on the y-axis; Outside air temperature is measured on the x-axis; Clusters are represented by the color of the points, which each represent a day; Note – The axes remain between 0 and 1 as the data has been normalized

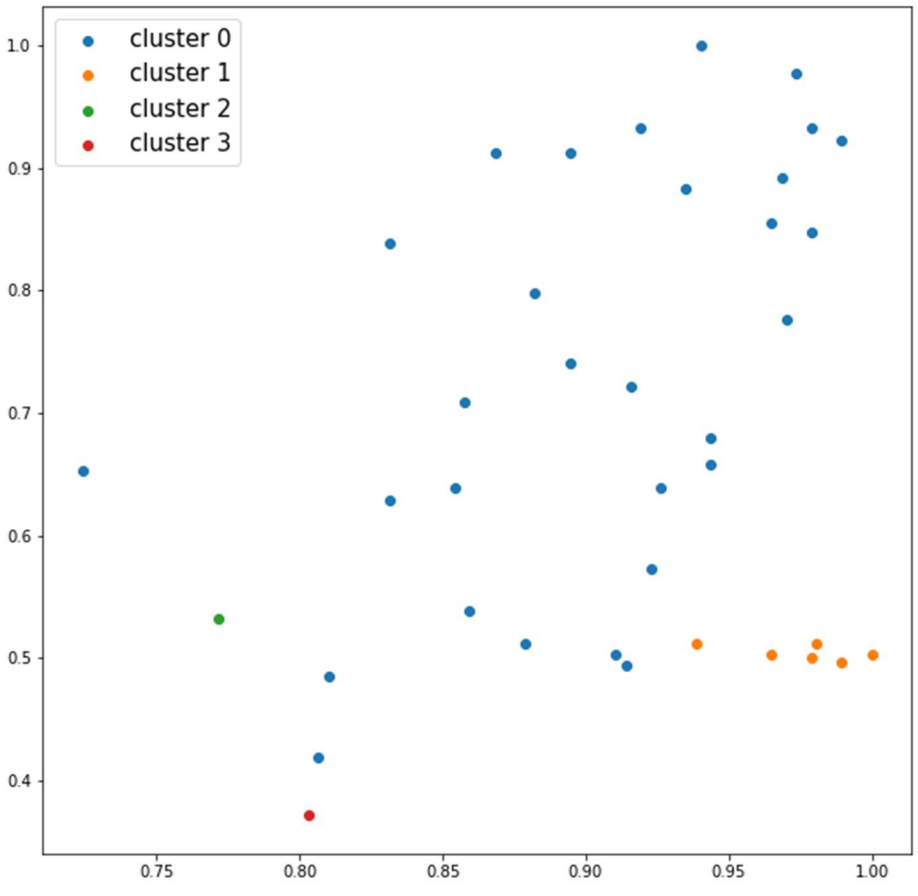
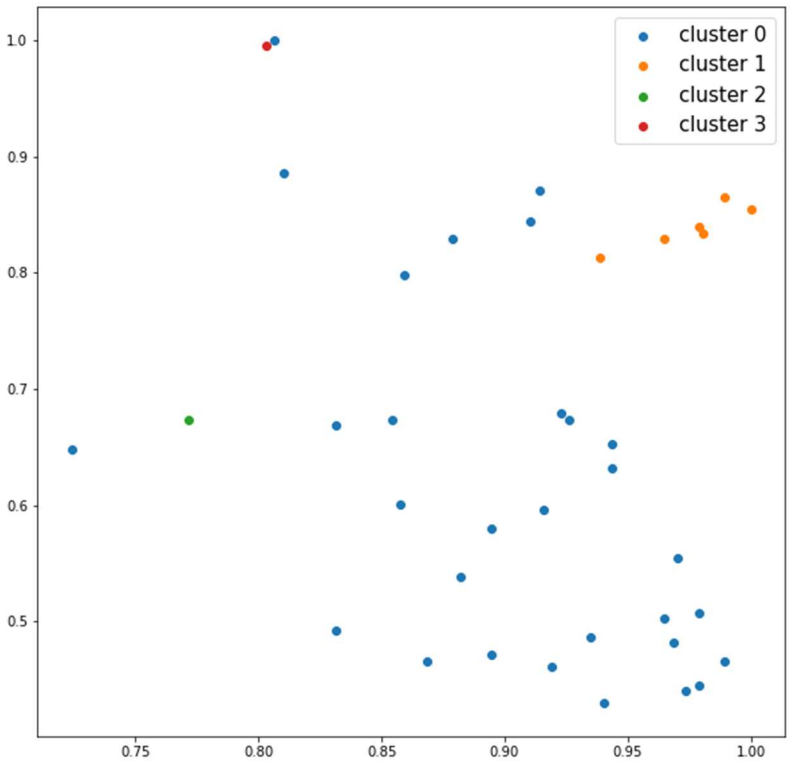


Figure 18 - kW/Ton against Outside Air Temperature

kW/Ton are measured on the y-axis; outdoor air temperature is measured on the x-axis; each point represents a day in the data set and the color of the point indicates the assigned cluster



From Figure 17, a conclusion can be reached regarding the COP measurements for the clustered data points in yellow. Simply stated, the COP represents the heat removed by the device and thus a higher COP is better when the device is on. The yellow data points from October (pre-treatment) manifest a low COP even when the temperature outside is measured near it's highest i.e. the device is not succeeding in removing large amounts of heat at high temperatures pre-treatment. Post-treatment the COP increases dramatically for the majority of the data points, with the exception of a handful of outliers. This means that the algorithm proves the effectiveness and confirms the expected positive impact that PowerTron's treatment has on the device. The effect is sure enough that even an algorithm that is fed unlabeled data picks up on the trend.

Figure 18 manifests a similar proof of expected treatment outcomes in the kW/Ton measurement. If a treatment were intended to improve the efficiency of a device, one would expect the amount of electricity used to supply a given amount of cooling to decrease. This is seen as the kW/Ton for those pre-treatment October days is clustered in the upper right hand of Figure 18 where the grouping of cluster 1, yellow dots are located. This indicates the majority of data post-treatment shows less energy being used to provide the same amount cooling, despite high temperature readings. The algorithm, despite not knowing which days were pre and post-treatment, has identified the difference in pre and post-treatment data and once again proves the efficacy of PowerTron's PermaFrost treatment on the Macy's unit.
An existing use of such findings is to use the data in advertising or marketing the treatment to prospective clients. This is evident from a high-level exploration, but also overlooks certain potential

uses for this method in other scenarios. For instance, this method could be used to examine source measurements (e.g. the wet bulb measurement and temp 1 & 2) and then compare the resulting cluster to the expectations set in the laws of physics and defined by the psychrometric chart. Given the nature of physical phenomenon, one would expect the relationships between certain source readings to fall within a particular cluster. If, when actively feeding the algorithm data day to day, the data begins to shift enough that it eventually triggers the creation of a new cluster by the algorithm, it could possibly point to a future mechanical failure or problem.

With limited success comes the need for further exploration of the algorithm's performance on different data sets and modified algorithm settings to shift the model and develop a deeper understanding. This initial examination deemed the algorithm to be successful enough that teams could explore the possibilities more thoroughly in the future.

Archetypal Analysis

Archetype Background

Archetypal Analysis (AA) is a tool for the study of large data sets, used in both data compression, and the analysis of spatio-temporal patterns in data (Cutler and Stone, 1997; Stone and Cutler, 1996; Bauckhage, 2014; Vinue et al., 2015). Recently, with the advent of greater computational power, it has been used in a variety of applications in the analysis of "Big Data". For instance, it has been used to analyze weather, climate and precipitation patterns (Hannachi and Trendafilov, 2017; Steinschneider and Lall, 2015; Su et al., 2017). A probabilistic framework for archetypes is developed in (Seth and Eugster, 2016). Its application to machine learning appears in Mørup and Hansen (2012). It is also used in biomedical and industrial engineering (Epifanio et al., 2013; Thøgersen et al., 2013), and in the analysis of terrorist events (Lundberg, 2019).

Archetypal Analysis was introduced by Cutler and Breiman in 1994 as variant of principal component analysis (PCA) that could capture 'archetypal patterns' in the data (Cutler and Breiman, 1994). Through AA, each time-based observation is represented as a convex combination of a limited number of points, called 'archetypes' or 'pure types', which may or may not be observed. These influential data points best describe the exterior surface of the original data set, and as convex combinations of the data points themselves, resemble the observations.

AA provides a number of advantages over commonly used techniques for data compression and clustering, such as Principal Component Analysis, or PCA (Pearson, 1901; Abdi and Williams, 2010) or k-mean clustering (MacQueen, 1967). PCA can lead to a complex representation of the data and is restricted by orthogonality, so meaningful features may not be discovered. Clustering approaches provide easy interpretation, but tend to lack modeling flexibility, with each observation grouped in only one cluster, no in between or intermediate groupings are allowed. In contrast, with AA, each observed data point is either classified to its closest archetype (single), or it is associated with two or more archetypes (mixed). Therefore, AA combines the virtue of both methods; it is easy to interpret and by allowing intermediates, provides more flexibility than clustering.

Mathematical Formulation

Appendix A shows the mathematical formulation for the archetypes in detail.

Results

The team applied archetypal analysis to the summary data set produced by Javier Perez-Alvaro, where pre-treatment is described in the hierarchical clustering section. Each data point represents one “cooling day” and contains some number of summary measurements for that day. To test the method, we combined the October 2019 data set with the May 2020 set, to see if archetypal analysis could distinguish pre- and post- treatment operation of the machine. We chose a subset of the full summary data set, considering only the median values, and pruning variables that have a direct and obvious relation with each other. For instance, the max OAtemp and the median OAtemp are plotted against each other in Figure 19. A linear regression shows that they are the same variable, up to a vertical shift. Therefore, one can be excluded without loss of generality. We chose to retain the maximum daily temperature. Similarly, EER and COP are also linearly related, as they are measurements of the same thing, in different units. See Figure 20. We chose the median COP for our data set. We then selected Kw/ton, which is inversely related to COP, generally speaking, but variations of this dependence could indicate varying operating conditions. Also included are median total capacity in BTU tons when compressor 1 is on, and when compressors 1 and 2 are on. Finally, the total time that compressor 1 and compressor 2 are on during the day were included. Compressor 3 and 4 are not always turned on each day, subject to control by the machine operator, but could be included in another round of analysis, as the total capacity will be effected by this.

Figure 19 - Max OAtemp vs. Median OAtemp

A linear relationship is shown by the line in yellow and described by the formula listed in the upper left corner of the figure

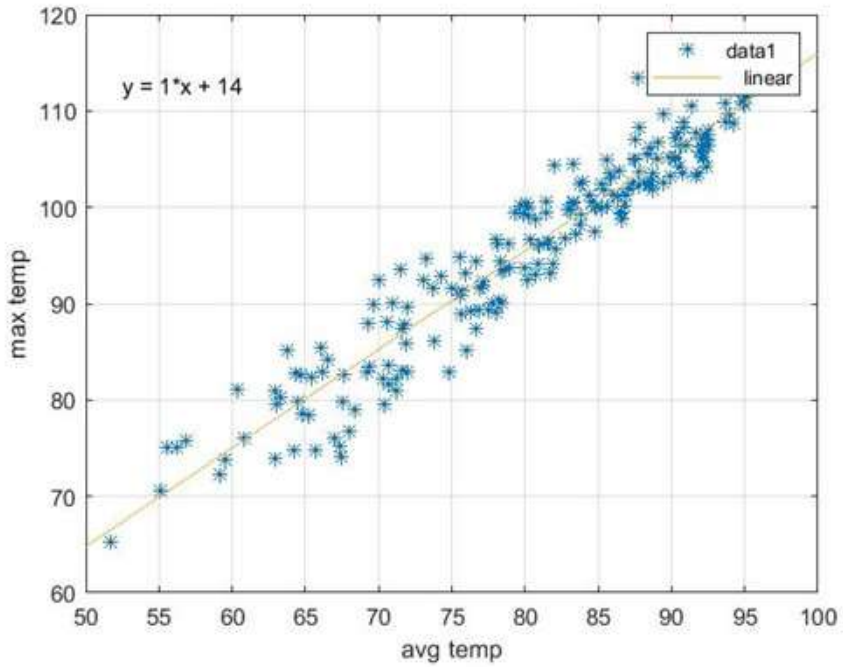
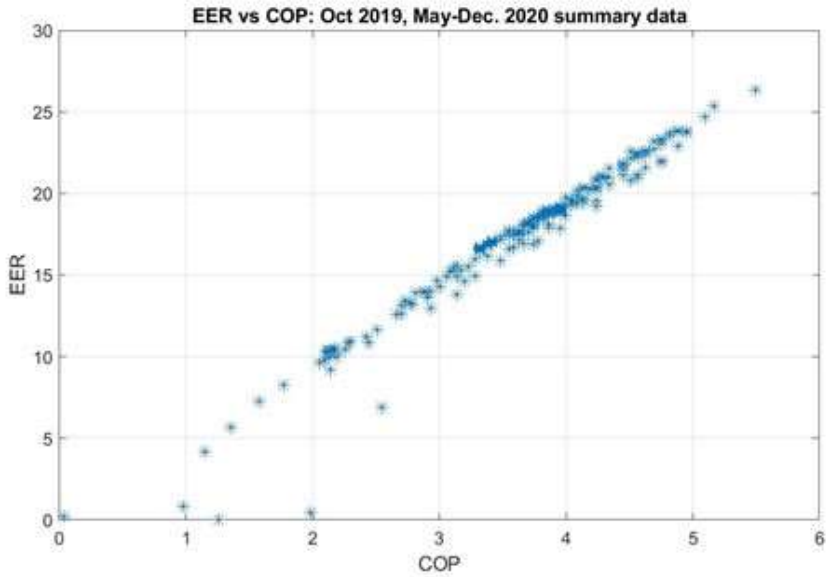


Figure 20 - Median EER plotted against Median COP

A linear relationship (with few outliers) is apparent.



In summary, the variables that make up each data point are:

1. Median Kw/ton - (kwtons)
2. Median COP - (med COP)
3. Median capacity in BTU tons when compressor 1 is on - (comp1Ton)
4. Median capacity in BTU tons when compressor 1 and 2 are on - (comp2Ton)
5. Max outside air temp in degrees F - (max temp)
6. Amount of time compressor 1 is on in seconds - (time1)
7. Amount of time compressor 2 is on in seconds - (time2)

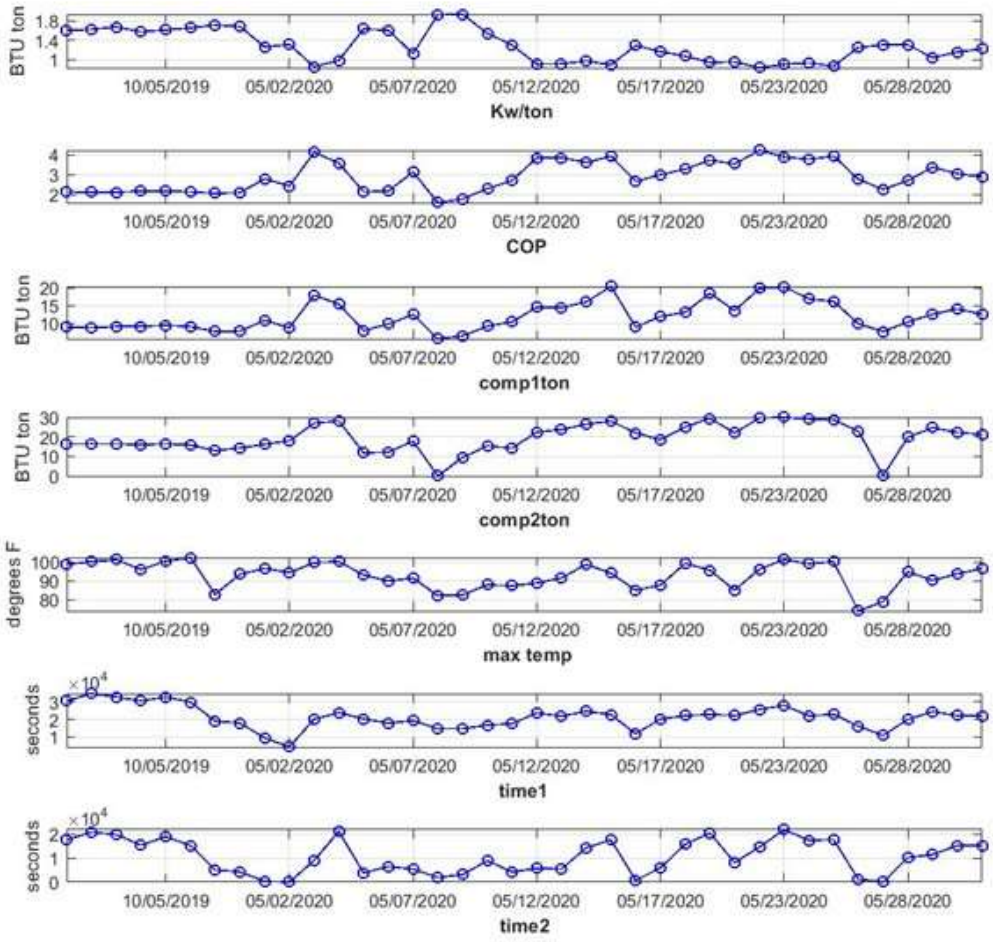
Dates of the 8 points from October 2019 are: 10/01-10/06 and 10/08-10/09.

Dates of the 30 points from May 2020 are: 05/01-05/30.

Figure 21 shows time series plots of the data.

Figure 21 - Time Series Plot of Each Variable

Dates are on the x-axis, units of the variable on the y-axis. Note that COP is unitless

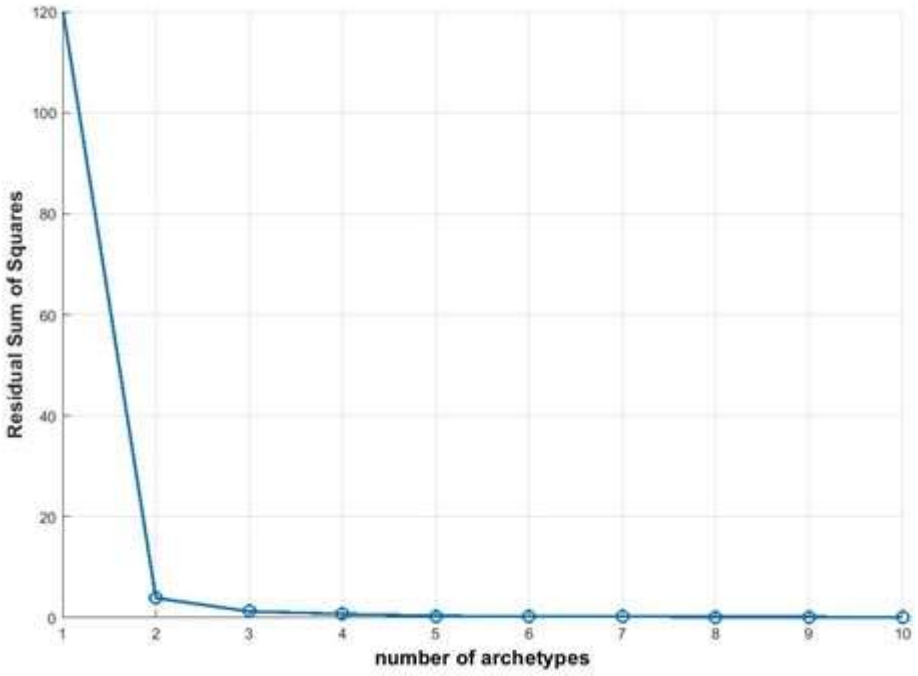


Prior to performing archetypal analysis, the 38 points are normalized in each dimension separately, by dividing each measurement by the maximum value of the measurement in the data set. Thus, the variables all ranges between 0 and 1, with fractions representing percentages, e.g., 0.5 is 50% of the max value achieved by that variable in the data set.

The first step in applying archetypal analysis is to compute the total error in representing the data in terms of the archetypes, in this case it is the residual sum of squares, or RSS. The RSS is computed for each set of archetypes and decreases as a larger number of archetypes is taken. See Figure 22.

Figure 22 - Residual Sum of Squares (RSS) vs. Number of Archetypes for the Oct2019/May2020 Data Set

The algorithm creates a set of N archetypes sequentially, each accompanied with the total error (RSS) in representing the data with that set of archetypes



With only one archetype computed the error is roughly 118.5, but drops quickly when a two archetype set is computed. We chose to truncate at 3 archetypes; adding any more does not significantly affect the accuracy of the data reconstruction. Archetypal analysis thus reduces the data set from 7 dimensions to 3, and the archetypes themselves are representative of typical "cooling days". A bar chart of the variables in each archetype is shown in figure 23. All three have similar large max temperatures, but archetype 2 is a more efficient day, as the median COP is larger and the Kw/ton is smaller for it than the other two. Also, compressors 1 and 2 have large capacity output, and both are on for a similar amount of time. Archetypes 1 and 3 are less efficient days, with higher Kw/ton and lower COP. Archetype 1 represents a day where compressor 2 is not turned on, and archetype 3 will capture less efficient days when both compressors are on. Each data point can be reconstructed by a convex combination of the 3 archetypes, and the mixture coefficients (the alphas) for each is plotted in figure 24.

Figure 23 - Bar Chart Representation of the 3 Archetype Set

Each archetype is like a data point from the set and made up of magnitudes of each of the seven measurement variables (betas).

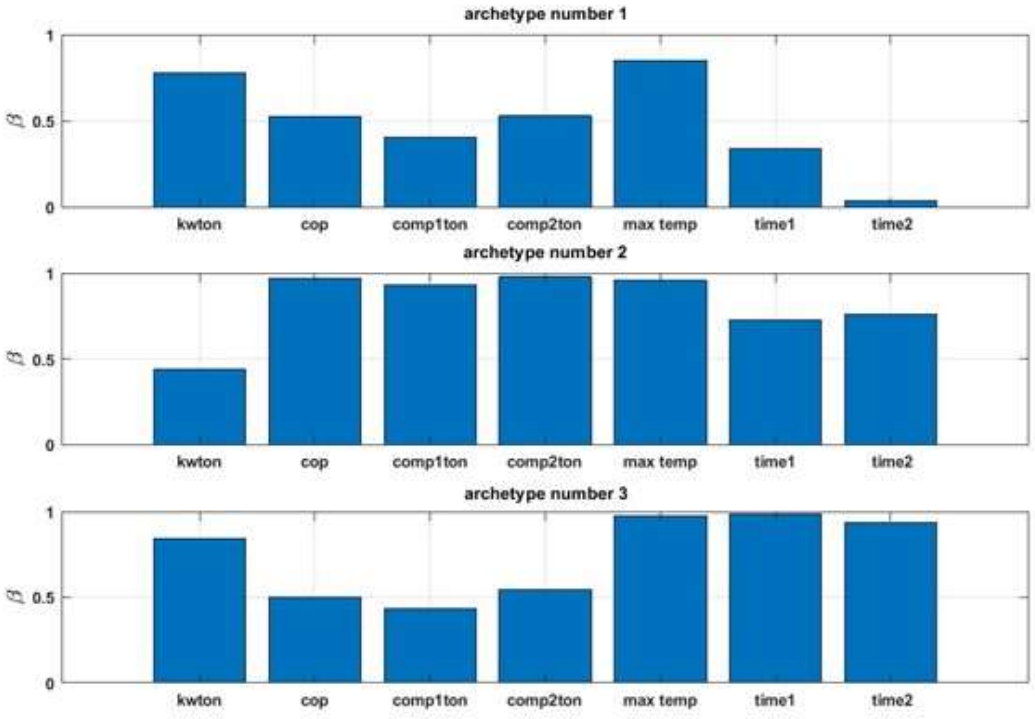
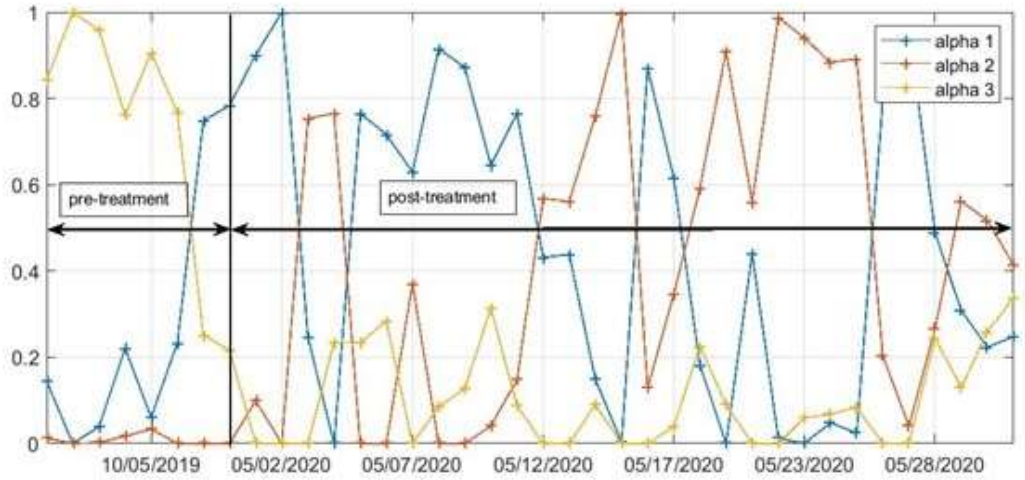


Figure 24 - Mixture Coefficients (alphas) for each data point, 1-38

The vertical line runs through Oct. 8th, the last day of the Oct. 2019 data set. From there on, the data points (days) are in May. Thus, the line divides pre-treatment days and post-treatment days

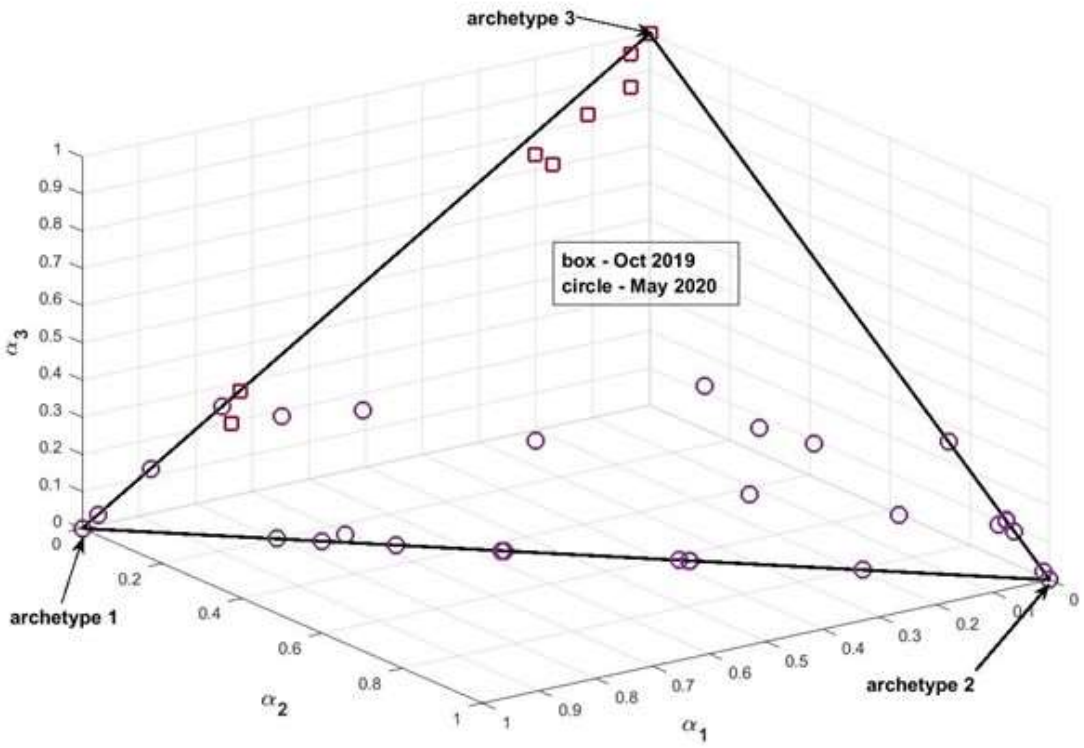


The mixture coefficient time series in figure 24 illustrate that archetype 3, and then a mixture of 3 and 1, represent the pre-treatment days. This is consistent with the properties of these two archetypes, and both are less efficient than archetype 2. The difference is if compressor 2 is on or

not. May days are represented largely by archetypes 1 and 2, a more efficient archetype combined with another where compressor 2 is turned off. A 4-archetype set would most likely split up the behavior better, but would not generate a significant reduction in the reconstruction error.

Figure 25 - 3D Plot of Data Points as Represented by 3 Alpha's

The simplex is enclosed by the triangle. Red points are from May 2020, blue from October 2019.



From this, it is evident that the pre-treatments are less efficient than the post-treatments, as would be hoped. It coincides with the findings of PowerTron in its Macys PermaFrost NMR Confirmatory Test report (August 18, 2020).

Another way to represent the data is a 3D plot of the mixture coefficients. These will lie on a simplex (represented by the triangular face in Figure 25) because the sum of the alpha's for each data point will be 1, and the alpha's themselves range between 0 and 1. Thus, each data point (cooling day) is represented (approximately) by a convex combination of alpha's and can be plotted on a 3-D simplex in the case of 3 archetypes. The first 6 days of Oct. 2019 lie in a cluster near archetype 3. The last 2 days are closer to archetype 1. This archetype captures the lower temperature days on Oct. 7th and 8th (see figure 21). The data points in May lie near the simplex edge between archetype 1 and 2, and thus are well represented by a convex combination of the two. This allows for a variation in efficiency and whether or not compressor 2 is on.

Conclusion and Possible Use Case

We have shown that archetypal analysis applied the summary variables chosen can differentiate between pre and post treatment days. It also shows sensitivity to variation in outdoor temperature and whether compressor 2 is on. We speculate that it could capture a gradual change in operating conditions through observations of the alpha's. If, over time, the data point drift from one condition to another, the representation in terms of archetypes would shift from all in the first archetypal condition, to all in the other, with points in between lying on the simplex boundary between the two. Another possible use would be the detection of anomalies. First, a set of archetypes representing the usual operation of the machine is created, using a training set of the summary variables. Additional days could be tested by projecting their data points onto the archetypal set (an optimization problem with a nonlinear constraint). If the error jumps up for the new data point, it could be anomalous. The daily calculation on board the machine would be the creation of the summary variables, and a projection onto the archetypal set. If an anomaly is detected, further classification that day's variables could be done to identify the source of the anomaly.

Lack of Test Data: An attempt to manually produce anomaly ridden data sets.

A theme throughout the testing of various methods was the lack of annotated data that would allow the team to test each method against known anomalies in the data stemming from mechanical failure. As such, an attempt was made to manually produce a training set of data and then plant known anomalies or failures within the set.

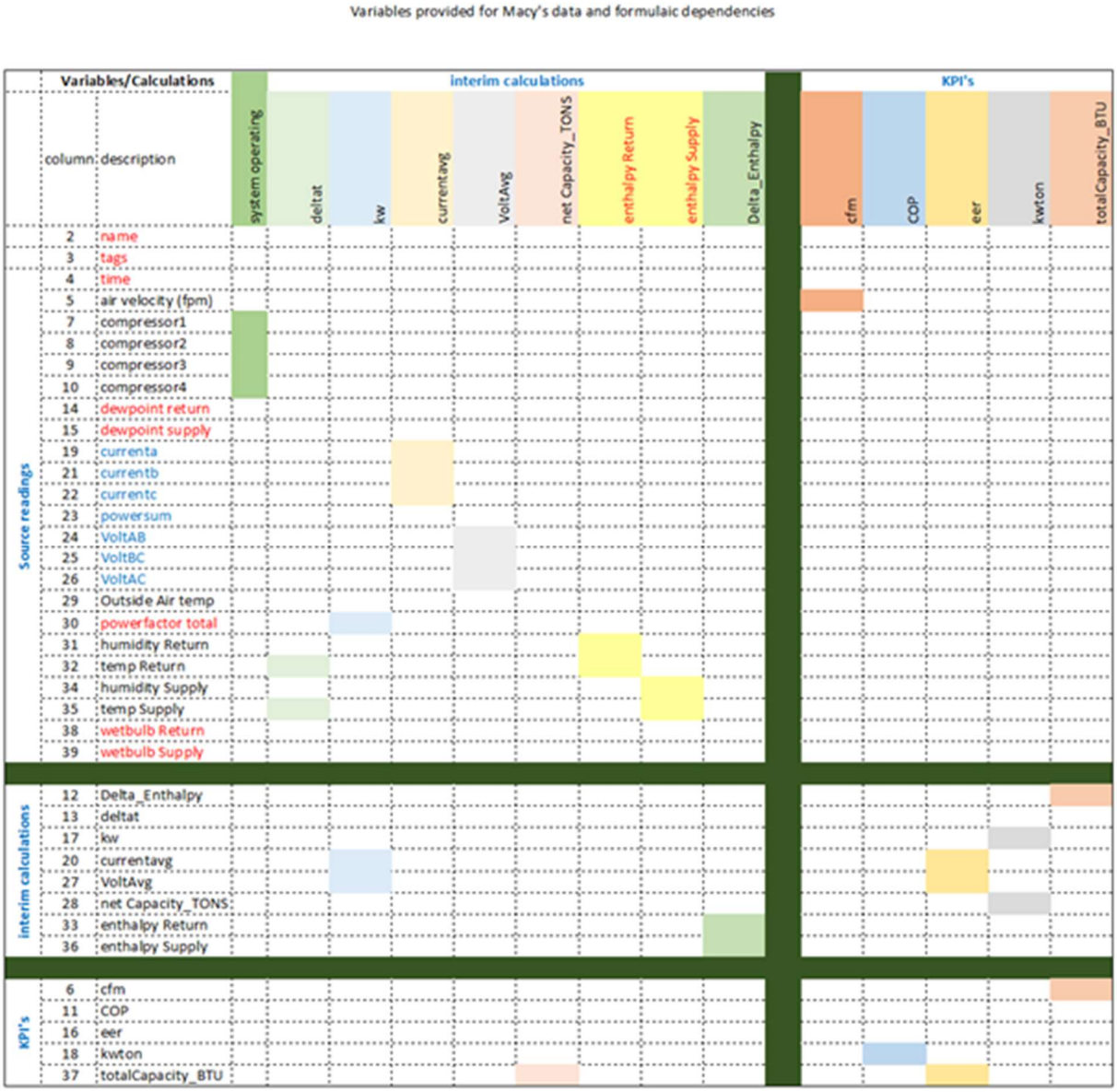
The following section describes the methodology used to develop the simulated set for documentation purposes only. The results from testing the methods described in previous sections on the simulated set were inconclusive. Further examination would be needed to determine the failure of the simulated data produced and modify the method accordingly.

Basis for Manual Calculations

From 25 fields of source data, 12 variables were used to produce 8 interim calculations, which then were used to calculate 5 key performance indicators related to the efficiency of the device. Manual replication of calculations was achieved via exploration of the relationships between all measurements. Figure 26 outlines the relationship between all variables and was also included in the team's milestone 1 report.

Figure 26 - Matrix Identifying Relationships Between Variables

Variables marked in red font were not used in the calculation of the efficiency KPIs



Within the source data for Macys, 15 months of complete or partial data was presented, captured in 1 second intervals. The result of this measurement frequency was an extremely large dataset only representative of a single year of data potentially impacted by seasonal weather variances.

Methodology for Producing Simulated Data

After examining and constructing the above table representing the relationship between source and calculated variables, development of simulated data began. Initially to ensure accurate construction of a dataset, the period and timing of source data, average relative humidity and high/low temperatures for Denton, Texas were gathered and used to create relevant values for the 15 month period in the source data. Average monthly low temperature was set for 6:00 AM on the 15th day of

a given month, while relevant high temperature was set for 6:00 PM of same day. From these anchor points, outside air temperature was linearly distributed to all 3-minute intervals represented between these monthly centers, with ratable temperature increase from 6:00 AM to 6:00 PM, and ratable decrease from 6:00 PM to 6:00 AM. This sampling is depicted in figures 27, 28, and 29.

Figure 27

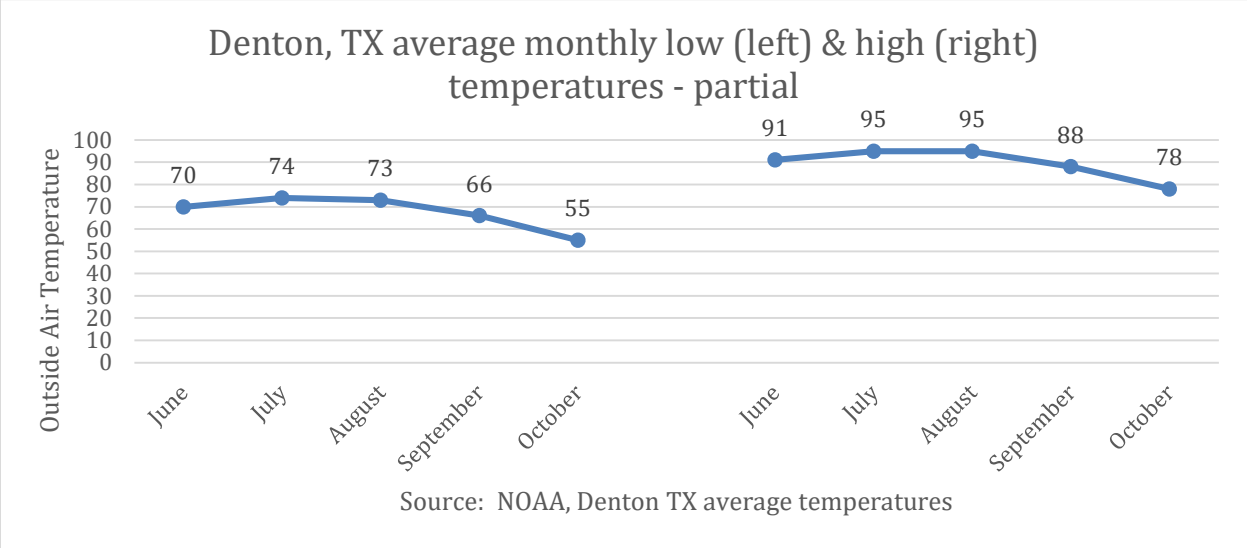


Figure 28

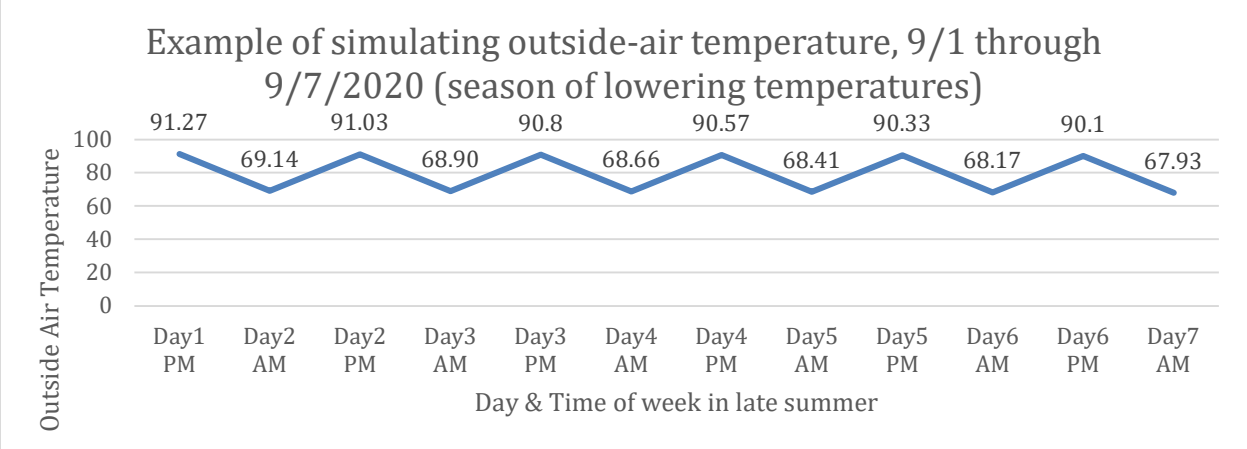
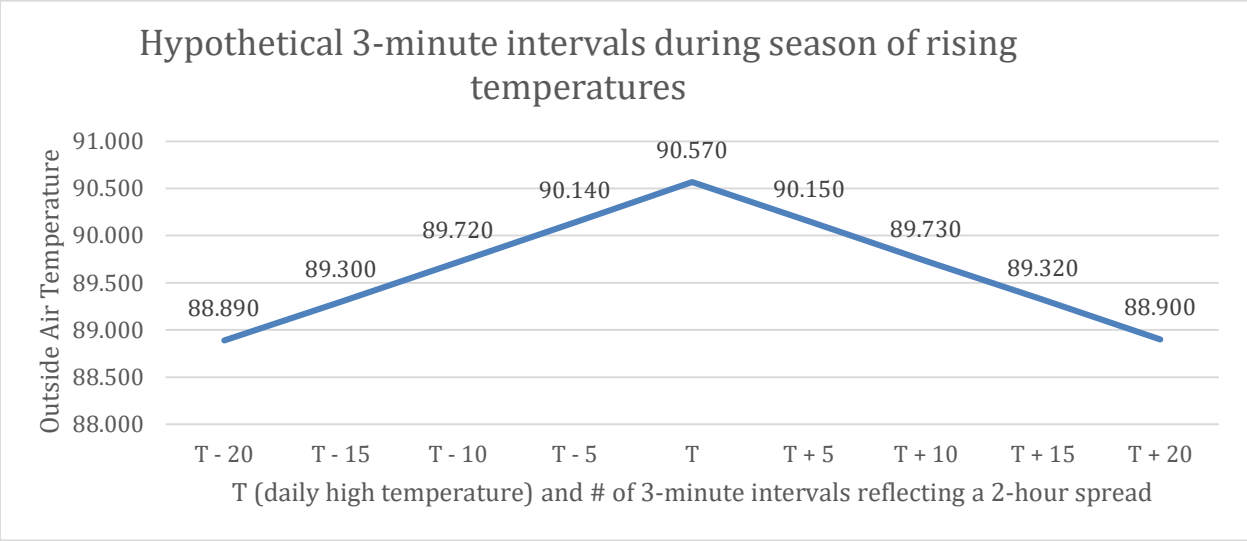


Figure 29

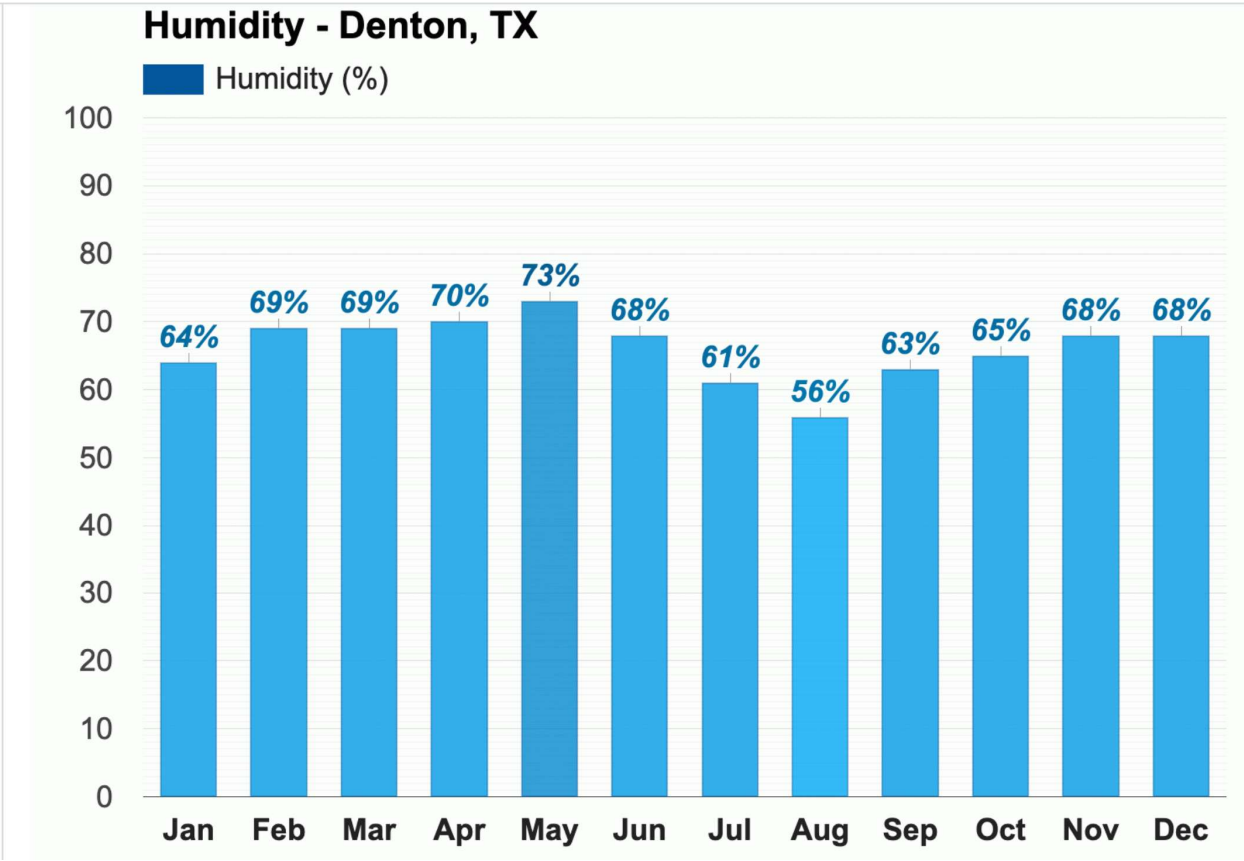


As is more apparent in this shorter time window, during a season of overall rising temperatures, the ratable drop in interval temperature following the high temperature point of the day is nominally less that were the ratable increases in interval temperature preceding the high temperature point of the day. The opposite follows during a season of lowering temperatures.

Humidity averages were applied evenly throughout the month identified to all 3-minute intervals contained in said month, depicted in Figure 30.[6]

[6] https://www.weather-us.com/en/texas-usa/denton-climate#humidity_relative

Figure 30



With outside air temperature (OAT) and humidity1 (H1) replaced by these values, a review of the average relationship between OAT and temp1, temp2 resulted in extrapolation of values for those additional columns; similarly, a review of H1 to humidity2 resulted in an extrapolation to populate humidity2.

Figure 31 represents the average results for manual calculations simulating the Macy's dataset.

Figure 31

Sample Average Macy's readings

2020 month	Avg OS air	Avg Hum1	Avg Temp1	Avg Hum2	Avg Temp2
Feb	74.21	37.75	78.83	48.68	69.3
Sep	83.44	53.95	80.55	73.18	68.62
Dec	72.74	22.74	76.34	29.25	68.04
composite	76.79667	38.14667	78.573333	50.37	68.653333
Feb			1.0622558	1.289536	0.9338364
Sep			0.9653643	1.356441	0.8223873
Dec			1.0494913	1.28628	0.9353863
factors to apply to simulated data			1.0231347	1.32043	0.8939624

A histogram of the OAT (Figure 32) results into bins provided a reasonable approach for any given 3-minute “on” interval to identify the number of compressors “active” (identified with a value = 1). A review of the source data velocities, coupled with the number of compressors active provided a means to assign a relevant velocity measure, with variance allowed for based on differing humidity levels by month.

Figure 32

Histogram of Simulated temperature readings

Temp Bin	Frequency
40	5651
50	24835
60	34546
70	31818
80	38209
90	26476
100	4895
total readings	166430

Resulting compressor count and velocity calculation:

compressor 1 active as per Macy’s actual data: 600 * humidity1

If OAT >= 78.5 = 2 compressors: 600 * humidity1 * 1.5

If OAT >= 85.5 = 3 compressors: 600 * humidity1 * 2.0

If OAT > 90 = 4 compressors: 600 * humidity1 * 2.5

The remaining 6 source data variables (current and voltage readings) were taken from the original source data as no significant anomalies were detected.

With these 12 variables for each 3-minute interval (166,430 of which 39,853 were compressor active), manual calculations of all interim and final KPI’s were performed and captured, leading to a 38-field simulation data set.

A random sample of 8,000 active records was then subjected to a process of additional calculation (where OAT was arbitrarily increased by 15 degrees for calculation purposes only), with the resulting KPI’s being labeled as anomalies. An additional field was created to label each record as a “1” (8,000 anomaly records) or a “0” (31,853 non-anomaly records).

The simulation model for calculating anomaly values in the KPI’s is relatively easy to adjust and other methods to calculate the anomalies may be useful for building a machine learning “train/test” model that could be later applied to the actual Macy’s data.

Conclusion on Simulated Data Set

When MAD and Random Forest were applied to the labeled simulated data set, unfortunately results produced were inconclusive. The manually created ‘anomalies’ were not identified in a consistent manner by either algorithm and further exploration via modifications to the simulation set were not possible due to time constraints.

Thus, the end conclusion on the manufactured anomalies and the simulated dataset was that the creation of this dataset would need to be either further explored and tested or abandoned. The simplest resolve would be to ensure real annotated data is collected as opposed to attempting to create a simulated set using the methodology described above. The limitation of the requirement for additional, labeled data to further examine methodologies explored was not resolved with this approach. Therefore, the need for labeled datasets for future exploration of methods stands.

Overall Limitations, Variables to Measure, and Conclusion

Limitations

The standing limitation for the testing and exploration of all methodologies for anomaly detection was the lack of labeled data. The need for an understanding of real-life decisions made by building engineers, device failures, and more would aid in assessing the actual success of certain algorithms, especially those that function as supervised machine learning models.

Time also proved to be a limitation, as the team had to first explore the data and possibilities for analysis as well as understand the HVAC process for interpretation of results. If data with notes, or even a log made by the building engineer with dates and times included, was produced, it would be possible to execute further examination of outliers or provide meaning to the clusters that an algorithm creates. Without this understanding, results bear success in a superficial manner that would benefit greatly from an on-the-ground understanding of the device's behavior and potential factors impacting this behavior.

Additional Variables to Actively Measure

PowerTron expressed explicit interest in understanding whether actively taking additional measurements would aid in future explorations. The following bulleted list represents possible items to measure actively that would add value to the datasets currently being produced by PowerTron's sensor technology:

- Amount of time daily that each compressor is spent on
- MAD for a selected increment (daily or monthly)
- Maximum and Minimum daily values for each variable
- Daily averages and medians for all variables
- Any record of mechanical failure or sensor failure
- Pressure (needing additional sensor equipment)

Conclusion

While the production of simulated data did not produce any additional results, preliminary results seen in the exploration of MAD, Random Forest, agglomerative hierarchical clustering, and archetypal analysis were promising and presented room for additional exploration.

Possible use cases for such algorithms range from the ability to notify individuals of a breach in threshold in real time to the ability to examine the impact of usage decisions and changes to a

device within the framework of historical device behavior. Several methods left space for deeper exploration to potential develop a method for predictive mechanical failure, however limitations mentioned would need to be addressed prior to implementing such exploration.

Successful application of a variety of analysis methods represents a true positive outcome of this project and opens the door to many possibilities in the world of analyzing HVAC sensor data and applying it to improving current business offerings as well as the potential for offering an entirely new service to existing customers if sensors were maintained post-treatment.

Appendix A. Mathematical Formulation of Archetypes

Mathematical Formulation

Consider an $m \times n$ matrix $\mathbf{X}$, where n is the number of observations in the data set. AA decomposes the spatiotemporal variability of $\mathbf{X}$ in a similar way as the PCA but with the following underlying constraints. Given a specified value for k, AA aims to identify m-dimensional vectors $\mathbf{z_1}, \cdots, \mathbf{z_k}$ that best describe k characteristic patterns, or archetypes, in the original dataset, such that data can be represented as convex combinations (i.e., linear combinations with nonnegative coefficients that sums to unity) of these archetypal patterns:

$$\mathbf{z}_j = \sum_{i=1}^{n} \beta_{ij} \mathbf{x}_i, \quad \beta_{ij} > 0 \ \ \& \ \ \sum_{i=1}^{n} \beta_{ij} = 1.$$

n-dimensional vector β_j contains the convex weights for the jth archetype across all observations.

Here, the convex weights, sometimes referred to as mixture coefficients, α_{ji} with $j = 1, \cdots, k$ range from 0 to 1 and reconstruct the ith observation across the k archetypes. The $k \times n$ matrix of all such weights is given by $\mathbf{A} = \{\alpha_1, \cdots, \alpha_n\}$. α_j can be viewed as the projection of the original data $\mathbf{X}$ onto the jth archetype $\mathbf{z}_j$, in the same way as PC scores do in PCA. The α_js are time series that determine how much of each archetype is used in every data point. Although the columns of $\mathbf{B}$ contain the time series of length n, we consider $\mathbf{A}$ to explore temporal variability of the archetypal patterns of $\mathbf{X}$. The reason for that is the time series columns of matrix $\mathbf{B}$ describe which time steps (e.g., days) are characterized best by one particular pure archetypal pattern and do not represent how dominant each archetype is in any given time step (e.g., day).

The $n \times k$ matrix of all such weights is given by $\mathbf{B} = \{\beta_1, \cdots, \beta_k\}$. Culter and Breiman show that each the archetypes are associated with real observations such that they are either convex combinations of the original observations or actual observations. This clearly facilitates interpretation compared to methods such as PCA. All observations can be approximated as a convex combination of the archetypes:

$$\hat{\mathbf{x}}_i = \sum_{j=1}^{k} \alpha_{ji} \mathbf{z}_j, \quad \alpha_{ji} > 0 \ \ \& \ \ \sum_{j=1}^{k} \alpha_{ji} = 1.$$

The $m \times k$ matrix $\mathbf{Z}$ of k archetypes is defined by the matrix factorization problem:

$$\min_{A,B} \|\mathbf{X} - \mathbf{XBA}\|,$$

where $\mathbf{Z} = \mathbf{XB}$. $RSS = \|\mathbf{X} - \mathbf{XBA}\|$ is residual sum of square errors and $\|.\|$ is the spectral norm. AA seeks to find k m-dimensional archetypes such that the loss function is minimized. This approach is described in more details in Cutler and Breiman. In a nutshell, AA uses a convex least-squares method (CLSM) to estimate the coefficient α_{ji}, subject to the constraints for given initial values of β_{ij}. Then it finds the best β_{ij} using CLSM, using the α_{ji}. This process iterates until RSS fails to improve. AA will find local minimums, not necessarily the global minimum of RSS, hence several starting values to insure a global solution are recommended. Furthermore, there is no universal method for optimal value of k. One commonly used approach is the "elbow" criteria, where a good value of k is selected when RSS fails to improve. This value can be determined by finding an elbow in the relationships between RSS and k using a screeplot. In this study, the algorithm is implemented using the a Principal Convex Hull Analysis (PCHA) package created by Morten Morup in Matlab.

Appendix B. Archetypal Analysis References

Bauckhage, C., 2014: A note on archetypal analysis and the approximation of convex hulls, arxiv:1410.0642
Cutler, A. and L. Breiman, 1994: Archetypal analysis. Technometrics, 36(4), 338–347.

Cutler, A. and E. Stone, 1997: Moving archetypes. Physica D: Nonlinear Phenomena, 107(1), 1 –16.

Epifanio, I., G. Vinue, and S. Alemany, 2013: Archetypal analysis: Contributions for estimating boundary cases in multivariate accommodation problem. Computers & Industrial Engineering, 64(3),
757 – 765.

Hannachi, A. and N. Trendafilov, 2017: Archetypal Analysis: Mining Weather and Climate Extremes.
Journal of Climate, 30(17), 6927–6944.

Lundberg, R., 2019: Archetypal terrorist events in the united states. Studies in Conflict & Terrorism, 42(9), 819–835.

Mørup, M. and L. K. Hansen, 2012: Archetypal analysis for machine learning and data mining. Neurocomputing, 80, 54 – 63, special Issue on Machine Learning for Signal Processing 2010.

Seth, S. and M. J. A. Eugster, 2016: Probabilistic archetypal analysis. Machine Learning, 102(1), 85–113.

Steinschneider, S. and U. Lall, 2015: Daily Precipitation and Tropical Moisture Exports across the Eastern United States: An Application of Archetypal Analysis to Identify Spatiotemporal Structure. Journal of Climate, 28(21), 8585–8602.

Stone, E. and A. Cutler, 1996: Archetypal analysis of spatio-temporal dynamics. Physica D: Nonlinear Phenomena, 90(3), 209 – 224.

Thøgersen, J. C., M. Mørup, S. Damkiær, S. Molin, and L. Jelsbak, 2013: Archetypal analysis of diverse pseudomonas aeruginosatranscriptomes reveals adaptation in cystic fibrosis airways. BMC Bioinformatics, 14(1), 279.

Vinue, G., I. Epifanio, and S. Alemany, 2015: Archetypoids: A new approach to define representative archetypal data. Computational Statistics & Data Analysis, 87, 102 – 115.